# PATENT ABSTRACTS OF JAPAN

(11)Publication number : **06-243113**
(43)Date of publication of application : **02.09.1994**

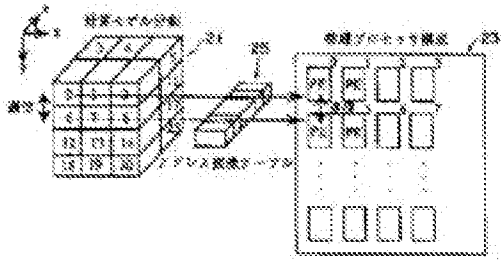
(51)Int.Cl. G06F 15/16

(21)Application number : **05-030971**
(22)Date of filing : **19.02.1993**

(71)Applicant : **FUJITSU LTD**
(72)Inventor : **SHIBATA KAZUYA**
**FUJISAKI MASAHIDE**
**KANAZAWA HIROYUKI**
**OKUDA MOTOI**

## (54) CALCULATION MODEL MAPPING METHOD FOR PARALLEL COMPUTER

(57)Abstract:
PURPOSE: To develop and operate the applications with no consciousness of the architecture of a parallel computer by securing the correspondence between the calculation units divided from a calculation model and the physical processors via an address conversion table.
CONSTITUTION: An N-dimensional calculation model 21 is divided into plural calculation units having the identification codes. Meanwhile a physical processor constitution 23 has a two-dimensional array. The identification codes are given to plural physical processors PE. The correspondence is secured between the calculation units and the processors PE by the identification codes and via an address conversion table 25. Under such conditions, those divided calculation units are optionally mapped to the processors PE via a user interface. Then a user can perform the communication among those processors by means of the identification codes of calculation units. Thus the work is attained in a user space and the mapping can be freely operated in accordance with the model 21.



(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号

特開平6－243113

(43)公開日　平成６年(1994)９月２日

| (51)Int.Cl.[5] | 識別記号 | 庁内整理番号 | ＦＩ | 技術表示箇所 |
|---|---|---|---|---|
| Ｇ０６Ｆ　15/16 | ３９０ | Ｔ　9190－5Ｌ | | |

審査請求　未請求　請求項の数31　ＯＬ　（全 28 頁）

(21)出願番号　特願平5－30971

(22)出願日　平成５年(1993)２月19日

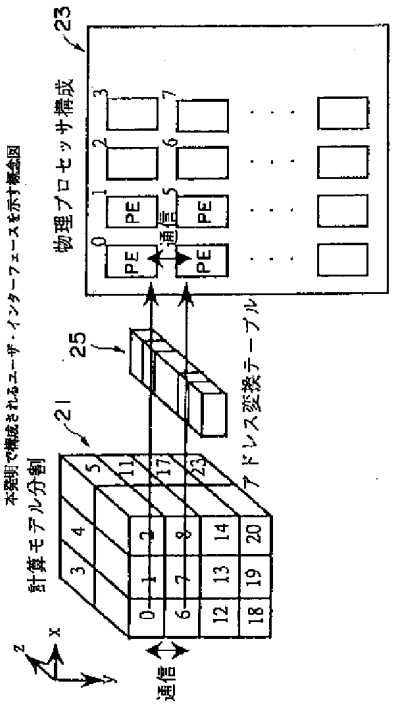
特許法第30条第１項適用申請有り　1992年８月20日　社団法人情報処理学会発行の「情報処理学会研究報告92－ＮＡ－42」に発表

(71)出願人　000005223
富士通株式会社
神奈川県川崎市中原区上小田中1015番地
(72)発明者　柴田　一哉
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内
(72)発明者　藤崎　正英
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内
(72)発明者　金澤　宏幸
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内
(74)代理人　弁理士　遠山　勉　（外１名）

最終頁に続く

(54)【発明の名称】　並列計算機における計算モデルのマッピング法

(57)【要約】
【目的】　ユーザ空間で並列計算機のアーキテクチャを意識することなくマッピングができ、かつ、高速なマッピングパターンを得ることを目的とする。
【構成】　ユーザが分割したＮ次元計算モデル２１の計算ユニットと、物理プロセッサ２３に識別符号をつけておき、その対応をアドレス変換テーブル２５でもつこれにユーザがアクセスするだけで、自由にマッピング操作ができるので、その後は、プロセッサ間通信はユーザが分割ユニットの識別符号を用いて行うことが出来る。



【特許請求の範囲】
【請求項１】　複数のプロセッサの間で相互にデータ及び情報を通信手段により転送することで各々のプロセッサが並列に処理を実行する並列計算機において、ユーザが演算処理しようとするＮ次元計算モデルを、前記複数のプロセッサで並列処理する方法であり、
前記Ｎ次元計算モデルを識別符号を有する複数の計算ユニットに分割するステップと、
前記分割した計算ユニットと各プロセッサとを対応づけるステップと、
でユーザ・インターフェースを形成し、
このユーザ・インターフェースを介して、分割した計算ユニットを、各プロセッサに任意にマッピングするステップを有し、
その後、プロセッサ間通信を分割ユニットの識別符号を用いて行うことを特徴とする並列計算機における計算モデルのマッピング法。
【請求項２】　請求項１において、通信コストの評価関数として、プロセッサ間の距離の総和Ｌを用い、前記マッピングは、この距離の総和Ｌが最小となるように行うことを特徴とする並列計算機における計算モデルのマッピング法。
【請求項３】　請求項２において、分割ユニットの隣接格子点Ｐｉ、Ｐｊが、それぞれ、２次元格子点上のある点（Ｘｉ、Ｙｉ）（Ｘｊ、Ｙｊ）にマッピングされるとき、
Ｐｉ、Ｐｊ間の距離１ｉｊは、｜Ｘｉ－Ｘｊ｜＋｜Ｙｉ－Ｙｊ｜と表され、
前記評価関数Ｌ＝Σ１ｉｊとし、これが最小となったときのマッピングを最適なものとして採用することを特徴とする並列計算機における計算モデルのマッピング法。
【請求項４】　請求項２または３において、初期マップ位置を決定した後、マップ位置を変化させ、長さの変化（前記距離の総和の変化）△Ｌを算出し、△Ｌ＜０のとき、前記マップ位置の変更を採用し、△Ｌ＜０でないとき、乱数Ｘ（０＜Ｘ＜１）を引き出してｅｘｐ（－△Ｌ／Ｔ）を演算し（Ｔは確立を決める目安となるパラメータ）、ｅｘｐ（－△Ｌ／Ｔ）＜Ｘのとき、マップ位置の変更を採用せず、ｅｘｐ（－△Ｌ／Ｔ）＜Ｘでないとき、マップ位置の変更を採用することとし、以上からの処理を、前記Ｔの値を減少させて繰り返すことを特徴とする並列計算機における計算モデルのマッピング法。
【請求項５】　請求項１において、前記マッピングは、３次元計算モデルを、ＸＹＺ方向の３方向の計算ユニットに立体的に分割し、それらを２次元に配列されたプロセッサにマッピングする場合であり、
前記３次元計算モデルをＺＸ平面でｎ個の計算ユニットに分割しておき、前記２次元において、Ｙ方向の各計算ユニットが正方形で隣合うようにし、ＺＸ方向の計算ユニットを一つ飛びに並べることを特徴とする並列計算機における計算モデルのマッピング法。
【請求項６】　請求項５において、（ａ）３次元計算モデルにおいて、Ｘ方向、Ｙ方向、Ｚ方向の計算単位をＶＣＸ、ＶＣＹ、ＶＣＺと定義し、Ｘ方向、Ｙ方向、Ｚ方向の分割数をＶＸ、ＶＹ、ＶＺと定義するとともに、
マッピング対象となる２次元物理プロセッサにおいて、Ｘ方向物理プロセッサ台数、Ｙ方向物理プロセッサ台数をそれぞれＰＣＸ、ＰＣＹと定義するとともに、Ｘ方向、Ｙ方向に分割した分割数をそれぞれＰＸ、ＰＹと定義して物理プロセッサをグループ化することを前提とし、（ｂ）３次元計算モデルをＶＸ＝ＰＣＸ／２、ＶＹ＝ＰＣＹ／２、ＶＺ＝０に分割し、この分割によって出来た直方体ユニットをメインユニットと呼び、各分割単位に通し番号（メインユニット番号）を付与し、さらにこのメインユニットをＺ軸方向に４つに分割し、この結果できたユニットをサブユニットと呼んで、各分割単位に通し番号（サブユニット番号）を付与し、
２次元物理プロセッサにおいて、ＰＸ＝ＰＣＸ／２、ＰＹ＝ＰＣＹ／２に分割してプロセッサをグループ化し、各分割単位に通し番号（グループ番号）を付与し、
（ｃ）３次元モデルのメインユニット番号と同一番号のグループ番号を有する物理プロセッサのグループにサブユニット番号０から３を、０と１、１と２、２と３、３と０とが隣接するように分配してマッピングすることを特徴とする並列計算機における計算モデルのマッピング法。
【請求項７】　請求項１において、前記マッピングは、３次元計算モデルを、ＸＹＺ方向の３方向の計算ユニットに立体的に分割し、それらを２次元に配列されたプロセッサにマッピングする場合であり、
前記３次元計算モデルをＺＸ平面でｎ個の計算ユニットに分割しておき、前記２次元において、各計算ユニットを順番に置き、ＺＸ方向が隣合いＹ方向が均等（ｎ個分離れる）に並べたことを特徴とする並列計算機における計算モデルのマッピング法。
【請求項８】　請求項７において、（ａ）３次元計算モデルにおいて、Ｘ方向、Ｙ方向、Ｚ方向の計算単位をＶＣＸ、ＶＣＹ、ＶＣＺと定義し、Ｘ方向、Ｙ方向、Ｚ方向の分割数をＶＸ、ＶＹ、ＶＺと定義するとともに、
マッピング対象となる２次元物理プロセッサにおいて、Ｘ方向物理プロセッサ台数、Ｙ方向物理プロセッサ台数をそれぞれＰＣＸ、ＰＣＹと定義するとともに、Ｘ方向、Ｙ方向に分割した分割数をそれぞれＰＸ、ＰＹと定義して物理プロセッサをグループ化することを前提とし、（ｂ）３次元計算モデルをＶＸ＝０、ＶＹ＝０、ＶＺ＝ＶＣＺに分割し、この分割によって出来た直方体ユニットをメインユニットと呼び、各分割単位に通し番号（メインユニット番号）を付与し、このメインユニットのＸ方向計算単位数、Ｙ方向計算単位数をｖｘ、ｖｙと

し、
２次元物理プロセッサにおいて、ＰＸ＝ＰＣＸ／ｖｘ、ＰＹ＝ＰＣＹ／ｖｙに分割してプロセッサをグループ化し、各分割単位に通し番号（グループ番号）を付与し、（ｃ）ｖｘ＝ＰＸ、ｖｙ＝ＰＹの位置に対応して、メインユニットを２次元物理プロセッサのＸ方向に通し番号順に配置し、そのライン上でのＸ方向の端部物理プロセッサに来たら、Ｙ方向へ移行し、今度はメインユニットを逆のＸ方向に通し番号順に配置し、再度Ｘ方向の端部物理プロセッサに来たら、Ｙ方向へ移行し、以後、以上の蛇行によるマッピングを繰り返すことを特徴とする並列計算機における計算モデルのマッピング法。

【請求項９】　請求項４の処理を並列処理することを特徴とする並列計算機における計算モデルのマッピング法。

【請求項１０】　請求項１において前記マッピングで得られた最適マッピング・パターンをデータベースに蓄積し、以後、このデータベースを参照して、計算モデルに適合する最適マッピングを選択し、並列処理をすることを特徴とする並列計算機における計算モデルのマッピング法。

【請求項１１】　複数のプロセッサの間で相互にデータ及び情報を通信手段により転送することで各々のプロセッサが並列に処理を実行する並列計算機において、ユーザが演算処理しようとするＮ次元計算モデルを、前記複数のプロセッサで並列処理するにあたり使用する通信ライブラリであり、
前記Ｎ次元計算モデルを識別符号を有する複数の計算ユニットに分割する分割ルーチンと、
分割した計算ユニットの識別符号と前記各プロセッサに付与された識別符号との対応関係をアドレス変換テーブル上に形成する管理ルーチンと、
分割した計算ユニットを、各プロセッサに任意にマッピングするマッピングルーチンと、
を備え、プロセッサ間通信を分割ユニットの識別符号を用いて行うことを特徴とする並列計算機用通信ライブラリ。

【請求項１２】　請求項１１において、前記マッピングルーチンにおける通信コストの評価関数として、プロセッサ間の距離の総和Ｌを用い、前記マッピングは、この距離の総和Ｌが最小となるように行うことを特徴とする並列計算機用通信ライブラリ。

【請求項１３】　請求項１２において、分割ユニットの隣接格子点Ｐｉ、Ｐｊが、それぞれ、２次元格子点上のある点（Ｘｉ、Ｙｉ）（Ｘｊ、Ｙｊ）にマッピングされるとき、
Ｐｉ、Ｐｊ間の距離ｌｉｊは、｜Ｘｉ－Ｘｊ｜＋｜Ｙｉ－Ｙｊ｜と表され、
前記評価関数Ｌ＝Σｌｉｊとし、これが最小となったときのマッピングを最適なものとして採用することを特徴とする並列計算機用通信ライブラリ。

【請求項１４】　請求項１２または１３において、初期マップ位置を決定した後、マップ位置を変化させ、
長さの変化（前記距離の総和の変化）△Ｌを算出し、△Ｌ＜０のとき、前記マップ位置の変更を採用し、△Ｌ＜０でないとき、乱数Ｘ（０＜Ｘ＜１）を引き出してｅｘｐ（－△Ｌ／Ｔ）を演算し（Ｔは確立を決める目安となるパラメータ）、ｅｘｐ（－△Ｌ／Ｔ）＜Ｘのとき、マップ位置の変更を採用せず、ｅｘｐ（－△Ｌ／Ｔ）＜Ｘでないとき、マップ位置の変更を採用することとし、以上からの処理を、前記Ｔの値を減少させて
繰り返すことを特徴とする並列計算機用通信ライブラリ。

【請求項１５】　請求項１１において、前記マッピングは、３次元計算モデルを、ＸＹＺ方向の３方向の計算ユニットに立体的に分割し、それらを２次元に配列されたプロセッサにマッピングする場合であり、
前記３次元計算モデルをＺＸ平面でｎ個の計算ユニットに分割しておき、前記２次元において、Ｙ方向の各計算ユニットが正方形で隣合うようにし、ＺＸ方向の計算ユニットを一つ飛びに並べることを特徴とする並列計算機用通信ライブラリ。

【請求項１６】　請求項１５において、（ａ）３次元計算モデルにおいて、Ｘ方向、Ｙ方向、Ｚ方向の計算単位をＶＣＸ、ＶＣＹ、ＶＣＺと定義し、Ｘ方向、Ｙ方向、Ｚ方向の分割数をＶＸ、ＶＹ、ＶＺと定義するとともに、
マッピング対象となる２次元物理プロセッサにおいて、Ｘ方向物理プロセッサ台数、Ｙ方向物理プロセッサ台数をそれぞれＰＣＸ、ＰＣＹと定義するとともに、Ｘ方向、Ｙ方向に分割した分割数をそれぞれＰＸ、ＰＹと定義して物理プロセッサをグループ化することを前提とし、（ｂ）３次元計算モデルをＶＸ＝ＰＣＸ／２、ＶＹ＝ＰＣＹ／２、ＶＺ＝０に分割し、この分割によって出来た直方体ユニットをメインユニットと呼び、各分割単位に通し番号（メインユニット番号）を付与し、さらにこのメインユニットをＺ軸方向に４つに分割し、この結果できたユニットをサブユニットと呼んで、各分割単位に通し番号（サブユニット番号）を付与し、
２次元物理プロセッサにおいて、ＰＸ＝ＰＣＸ／２、ＰＹ＝ＰＣＹ／２に分割してプロセッサをグループ化し、各分割単位に通し番号（グループ番号）を付与し、
（ｃ）３次元モデルのメインユニット番号と同一番号のグループ番号を有する物理プロセッサのグループにサブユニット番号０から３を、０と１、１と２、２と３、３と０とが隣接するように分配してマッピングすることを特徴とする並列計算機用通信ライブラリ。

【請求項１７】　請求項１１において、前記マッピングは、３次元計算モデルを、ＸＹＺ方向の３方向の計算ユニットに立体的に分割し、それらを２次元に配列された

プロセッサにマッピングする場合であり、
前記３次元計算モデルをＺＸ平面でｎ個の計算ユニットに分割しておき、前記２次元において、各計算ユニットを順番に置き、ＺＸ方向が隣合いＹ方向が均等（ｎ個分離れる）に並べたことを特徴とする並列計算機用通信ライブラリ。
【請求項１８】　請求項１７において、（ａ）３次元計算モデルにおいて、Ｘ方向、Ｙ方向、Ｚ方向の計算単位をＶＣＸ、ＶＣＹ、ＶＣＺと定義し、Ｘ方向、Ｙ方向、Ｚ方向の分割数をＶＸ、ＶＹ、ＶＺと定義するとともに、
マッピング対象となる２次元物理プロセッサにおいて、Ｘ方向物理プロセッサ台数、Ｙ方向物理プロセッサ台数をそれぞれＰＣＸ、ＰＣＹと定義するとともに、Ｘ方向、Ｙ方向に分割した分割数をそれぞれＰＸ、ＰＹと定義して物理プロセッサをグループ化することを前提とし、（ｂ）３次元計算モデルをＶＸ＝０、ＶＹ＝０、ＶＺ＝ＶＣＺに分割し、この分割によって出来た直方体ユニットをメインユニットと呼び、各分割単位に通し番号（メインユニット番号）を付与し、このメインユニットのＸ方向計算単位数、Ｙ方向計算単位数をｖｘ、ｖｙとし、
２次元物理プロセッサにおいて、ＰＸ＝ＰＣＸ／ｖｘ、ＰＹ＝ＰＣＹ／ｖｙに分割してプロセッサをグループ化し、各分割単位に通し番号（グループ番号）を付与し、（ｃ）ｖｘ＝ＰＸ、ｖｙ＝ＰＹの位置に対応して、メインユニットを２次元物理プロセッサのＸ方向に通し番号順に配置し、そのライン上でのＸ方向の端部物理プロセッサに来たら、Ｙ方向へ移行し、今度はメインユニットを逆のＸ方向に通し番号順に配置し、再度Ｘ方向の端部物理プロセッサに来たら、Ｙ方向へ移行し、以後、以上の蛇行によるマッピングを繰り返すことを特徴とする並列計算機用通信ライブラリ。
【請求項１９】　請求項１４の処理を並列処理することを特徴とする並列計算機用通信ライブラリ。
【請求項２０】　請求項１１において前記マッピングで得られた最適マッピング・パターンをデータベースに蓄積するルーチンを有し、以後、このデータベースを参照して、計算モデルに適合する最適マッピングを選択し、並列処理をすることを特徴とする並列計算機用通信ライブラリ。
【請求項２１】　複数のプロセッサの間で相互にデータ及び情報を通信手段により転送することで、ユーザが演算処理しようとするＮ次元計算モデルを、前記複数のプロセッサで並列処理する並列計算機において、
前記Ｎ次元計算モデルを識別符号を有する複数の計算ユニットに分割する計算モデル分割部と、
前記複数の各プロセッサに識別符号をつけておき、分割した計算ユニットと各プロセッサとの対応関係をアドレス変換テーブル上に形成する管理部と、
分割した計算ユニットを、各プロセッサに任意にマッピングするマッピング部と、を備え、
プロセッサ間通信を分割ユニットの識別符号を用いて行うことを特徴とする並列計算機。
【請求項２２】　請求項２１において、前記マッピング部は、通信コストの評価関数として、プロセッサ間の距離の総和Ｌを演算するプロセッサ間距離演算部と、距離の総和Ｌが最小となったときのマッピングを最適なものとして採用することを特徴とする並列計算機。
【請求項２３】　請求項２２において、前記マッピング部は、前記プロセッサ間距離演算部で得られた距離の総和Ｌが最小であるか否かを判定する判定部とを有し、距離の総和Ｌが最小となったと判定部が判定したときのマッピングを最適なものとして採用することを特徴とする並列計算機。
【請求項２４】　請求項２１において、分割ユニットの隣接格子点Ｐｉ、Ｐｊが、それぞれ、２次元格子点上のある点（Ｘｉ、Ｙｉ）（Ｘｊ、Ｙｊ）にマッピングされるとき、
Ｐｉ、Ｐｊ間の距離１ｉｊは、｜Ｘｉ－Ｘｊ｜＋｜Ｙｉ－Ｙｊ｜と表され、
前記評価関数Ｌ＝Σ１ｉｊとし、このＬが最小であるとされたときのマッピングを最適なものとして採用することを特徴とする並列計算機。
【請求項２５】　請求項２２または２３において、初期マップ位置を決定した後、マップ位置を変化させ、
長さの変化（前記距離の総和の変化）△Ｌを算出し、
△Ｌ＜０のとき、前記マップ位置の変更を採用し、△Ｌ＜０でないとき、乱数Ｘ（０＜Ｘ＜１）を引き出してｅｘｐ（－△Ｌ／Ｔ）を演算し（Ｔは確立を決める目安となるパラメータ）、ｅｘｐ（－△Ｌ／Ｔ）＜Ｘのとき、マップ位置の変更を採用せず、ｅｘｐ（－△Ｌ／Ｔ）＜Ｘでないとき、マップ位置の変更を採用することとし、以上からの処理を、前記Ｔの値を減少させて繰り返すことを特徴とする並列計算機。
【請求項２６】　請求項２１において、前記マッピング部は、３次元計算モデルを、ＸＹＺ方向の３方向の計算ユニットに立体的に分割し、それらを２次元に配列されたプロセッサにマッピングする場合であり、
前記３次元計算モデルをＺＸ平面でｎ個の計算ユニットに分割しておき、前記２次元において、Ｙ方向の各計算ユニットが正方形で隣合うようにし、ＺＸ方向の計算ユニットを一つ飛びに並べることを特徴とする並列計算機。
【請求項２７】　請求項２６において、（ａ）３次元計算モデルにおいて、Ｘ方向、Ｙ方向、Ｚ方向の計算単位をＶＣＸ、ＶＣＹ、ＶＣＺと定義し、Ｘ方向、Ｙ方向、Ｚ方向の分割数をＶＸ、ＶＹ、ＶＺと定義するとともに、
マッピング対象となる２次元物理プロセッサにおいて、

Ｘ方向物理プロセッサ台数、Ｙ方向物理プロセッサ台数をそれぞれＰＣＸ、ＰＣＹと定義するとともに、Ｘ方向、Ｙ方向に分割した分割数をそれぞれＰＸ、ＰＹと定義して物理プロセッサをグループ化することを前提とし、（ｂ）３次元計算モデルをＶＸ＝ＰＣＸ／２、ＶＹ＝ＰＣＹ／２、ＶＺ＝０に分割し、この分割によって出来た直方体ユニットをメインユニットと呼び、各分割単位に通し番号（メインユニット番号）を付与し、さらにこのメインユニットをＺ軸方向に４つに分割し、この結果できたユニットをサブユニットと呼んで、各分割単位に通し番号（サブユニット番号）を付与し、
２次元物理プロセッサにおいて、ＰＸ＝ＰＣＸ／２、ＰＹ＝ＰＣＹ／２に分割してプロセッサをグループ化し、各分割単位に通し番号（グループ番号）を付与し、
（ｃ）３次元モデルのメインユニット番号と同一番号のグループ番号を有する物理プロセッサのグループにサブユニット番号０から３を、０と１、１と２、２と３、３と０とが隣接するように分配してマッピングすることを特徴とする並列計算機。

【請求項２８】　請求項２１において、前記マッピング部は、３次元計算モデルを、ＸＹＺ方向の３方向の計算ユニットに立体的に分割し、それらを２次元に配列されたプロセッサにマッピングする場合であり、
前記３次元計算モデルをＺＸ平面でｎ個の計算ユニットに分割しておき、前記２次元において、各計算ユニットを順番に置き、ＺＸ方向が隣合いＹ方向が均等（ｎ個分離れる）に並べたことを特徴とする並列計算機。

【請求項２９】　請求項２８において、（ａ）３次元計算モデルにおいて、Ｘ方向、Ｙ方向、Ｚ方向の計算単位をＶＣＸ、ＶＣＹ、ＶＣＺと定義し、Ｘ方向、Ｙ方向、Ｚ方向の分割数をＶＸ、ＶＹ、ＶＺと定義するとともに、
マッピング対象となる２次元物理プロセッサにおいて、Ｘ方向物理プロセッサ台数、Ｙ方向物理プロセッサ台数をそれぞれＰＣＸ、ＰＣＹと定義するとともに、Ｘ方向、Ｙ方向に分割した分割数をそれぞれＰＸ、ＰＹと定義して物理プロセッサをグループ化することを前提とし、（ｂ）３次元計算モデルをＶＸ＝０、ＶＹ＝０、ＶＺ＝ＶＣＺに分割し、この分割によって出来た直方体ユニットをメインユニットと呼び、各分割単位に通し番号（メインユニット番号）を付与し、このメインユニットのＸ方向計算単位数、Ｙ方向計算単位数をｖｘ、ｖｙとし、
２次元物理プロセッサにおいて、ＰＸ＝ＰＣＸ／ｖｘ、ＰＹ＝ＰＣＹ／ｖｙに分割してプロセッサをグループ化し、各分割単位に通し番号（グループ番号）を付与し、（ｃ）ｖｘ＝ＰＸ、ｖｙ＝ＰＹの位置に対応して、メインユニットを２次元物理プロセッサのＸ方向に通し番号順に配置し、そのライン上でのＸ方向の端部物理プロセッサに来たら、Ｙ方向へ移行し、今度はメインユニットを逆のＸ方向に通し番号順に配置し、再度Ｘ方向の端部物理プロセッサに来たら、Ｙ方向へ移行し、以後、以上の蛇行によるマッピングを繰り返すことを特徴とする並列計算機。

【請求項３０】　請求項２５の処理を並列処理することを特徴とする並列計算機。

【請求項３１】　請求項２１において、さらにデータベース部を有し、前記マッピングで得られた最適マッピング・パターンをこのデータベース部に蓄積し、以後、このデータベースを参照して、計算モデルに適合する最適マッピングを選択し、並列処理をすることを特徴とする並列計算機。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は、並列計算機における関し、特に、並列計算機におけるマッピングを効率よく行い、通信の効率化を図る技術に関する。

【０００２】

【従来の技術】コンピュータシステムの高速化や大容量化が要求されるに伴って、複数のプロセッサに処理を分散させる分散処理技術が必要となってきた。

【０００３】そこで、複数のプロセッサが処理を並列的に行なう並列計算機が提供されている。この並列計算機では、各プロセッサが通信手段を介して互いに通信を行うことで各プロセッサが並列に動作し、複数のプロセッサ全体であるまとまった処理を実行する。これによれば、１つの仕事に対する処理の高速化を図ることができる。

【０００４】近年は、このような並列計算機の開発が進み、並列化を行う環境が次第に整い、実用的なアプリケーションが適応可能となってきた。しかし、並列化の過程で必要となる作業、例えばマッピングなどは未だ自動化されていない。特に、マッピングの際のデータ分割や、機能分割のアルゴリズム（以降、分割アルゴリズム）を考える時に、並列計算機のネットワーク・アーキテクチャの構造を考慮に入れなければならない。

【０００５】この場合、並列計算機のネットワーク・アーキテクチャの構造が様々であることから、効率のよいプログラムを開発するためには、一般ユーザが、マシンのアーキテクチャを十分に考慮する必要がある。

【０００６】このようにして完成した並列アプリケーション・プログラムでは、並列計算機のアーキテクチャが進歩した場合、再度、改良されたアーキテクチャに合わせてプログラムを変更しなければならない。

【０００７】従って、事実上は、並列アプリケーション・プログラムは並列計算機のアーキテクチャの変化に追いついて行けないであろう。これでは、適応性の高い並列アプリケーション・プログラムの蓄積は望めない。

【０００８】

【発明が解決しようとする課題】このような現状の環境

にもかかわらず、一部の先進ユーザは、アプリケーションの高速化を達成するために、アーキテクチャを十分に考慮して高速化を行っている。

【０００９】一方、並列計算機を普及させるには、一般ユーザに分割アルゴリズムを意識させないで、並列アプリケーションを開発させる環境が必要である。それには、使い勝手だけを考えれば、ユーザが自分のＮ次元計算モデルのイメージのままで並列アプリケーション・プログラムの開発が出来ればよい。また、アプリケーションの高速化という観点では、先進ユーザの計算モデルに対応したマッピングが必要になる。

【００１０】そこで、アプリケーションによく見られるデータ参照パターンについて、ユーザに使いやすくかつ、高速であるインターフェースが必要となる。本発明は、このような要請に基づきなされたもので、並列計算機において、ユーザが、並列計算機のアーキテクチャを意識せずに、アプリケーションの開発あるいは運用を図れるようにすることを課題とする。

【００１１】

【課題を解決するための手段】本発明は、前記課題を解決するため、複数のプロセッサの間で相互にデータ及び情報を通信手段により転送することで各々のプロセッサが並列に処理を実行する並列計算機において、以下の構成をとった。

【００１２】図１の原理図に示したように、並列計算機において、計算モデルを分割した計算ユニットと、並列計算機のアーキテクチャである複数の物理プロセッサとが、アドレス変換テーブルを介して、対応づけられる。

【００１３】図１で、２１は、ユーザが演算処理しようとするＮ次元計算モデルである。ここでは３次元モデルであるが、任意のＮ次元（Ｎ＝自然数）が可能である。Ｎ次元計算モデルは、識別符号を有する複数の計算ユニットに分割される。

【００１４】また、２３は、物理プロセッサ構成である。ここでは、２次元に配列されている。複数の物理プロセッサ（ＰＥ）には、識別符号が付与されている。２５は、分割された複数の計算ユニットと、各プロセッサとを、識別符号により対応づけるアドレス変換テーブルである。

【００１５】ユーザが計算させようとする計算モデルは様々であるが、適応する問題にはアプリケーション独特のデータ参照パターンが存在していて、さらに計算モデル構成が３次元以上の高次元であるものが多く、ネットワークアーキテクチャの構成と異なることが多い。前記問題を解決するためには、これを反映させる必要がある。

【００１６】これらを踏まえ、ユーザに、アーキテクチャを意識させず、計算モデルのイメージのままで作業させるためには、あたかも、物理プロセッサ構成がユーザの計算モデルと同じ構成をしているかのごとく、ユーザにイメージさせて、その空間（ユーザ空間）のイメージのままでプロセッサ間通信をさせるインターフェースを定義しなければならない。このようなインターフェースを実現するのが以上の構成である。

【００１７】そして、このユーザ・インターフェースを介して、分割した計算ユニットを、各プロセッサに任意にマッピングする。マッピングとは、計算モデルを、アプリケーションプログラム上で並列に計算できる計算領域の単位（ユニット）に分割したあとのプロセスであり、各ユニットを並列計算機上の、各プロセッサに割り当てることをいう。ここで、前記単位（ユニット）は、アプリケーションプログラムで並列に計算できる計算領域の最小単位ユニットでもよいし、この最小単位ユニットをいくつか集めたユニットブロックであってもよい。

【００１８】本発明は、方法として、あるいは、ライブラリ・プログラムとして、さらには、並列計算機自体としてとらえることが可能である。まず、方法としてとらえた場合、並列計算機において、ユーザが演算処理しようとするＮ次元計算モデルを、前記複数のプロセッサで並列処理する方法であり、前記Ｎ次元計算モデルを識別符号を有する複数の計算ユニットに分割するステップと、前記分割した計算ユニットと各プロセッサとを対応づけるステップとでユーザ・インタフェースを形成する。このインターフェースを介して、分割した計算ユニットを、各プロセッサに任意にマッピングするステップを有する。その後、プロセッサ間通信を分割ユニットの識別符号を用いて行う。

【００１９】この方法をライブラリ・プログラムとした場合、前記Ｎ次元計算モデルを識別符号を有する複数の計算ユニットに分割する分割ルーチンと、分割した計算ユニットの識別符号と前記各プロセッサに付与された識別符号との対応関係をアドレス変換テーブル上に形成する管理ルーチンと、分割した計算ユニットを、各プロセッサに任意にマッピングするマッピングルーチンと、を備え、プロセッサ間通信を分割ユニットの識別符号を用いて行うことを特徴とする並列計算機用通信ライブラリとして実現できる。これを、従来のライブラリ群に加えて、Ｎ次元サブルーチン・ライブラリとする。

【００２０】このようなライブラリの実行により、並列計算機において、前記Ｎ次元計算モデルを識別符号を有する複数の計算ユニットに分割する計算モデル分割部と、前記複数の各プロセッサに識別符号をつけておき、分割した計算ユニットと各プロセッサとの対応関係をアドレス変換テーブル上に形成する管理部と、分割した計算ユニットを、各プロセッサに任意にマッピングするマッピング部と、が実現される。

【００２１】以上において、計算モデルに対応した最適なマッピングを行う必要がある。最適なマッピングとは、アプリケーションのデータ参照パターンに対し、高速化という観点で最も効率的なプロセッサ間通信を実現

するものである。
【００２２】本来、ユーザ空間から、プロセッサ空間へのマッピング作業は、自動的に最適な形が選ばれるのが理想的である。しかし、アプリケーション独自のデータ参照パターンは、ユーザしか知らないもので、これまで多くの並列計算機システムでは、満足のいくマッピングを行ってはいなかった。また、最適なマッピング手法自体分からない部分が多い。そこで、本発明では、まず各通信パターンに対応した、最適なマッピング規則を見つけることに目標をおいた。そのためには、ユーザが自由にマッピング規則を操作できるものが必要になる。
【００２３】前記で説明した構成は、このような要請に応えるユーザ・インターフェースを提供する。マッピングに当たっては、通信の高速化の観点から、ネットワークアーキテクチャに加え、新たにアプリケーションのデータ参照関係を考慮する必要がある。そこで、マッピングの決定という問題を前述のように通信の高速化問題としてとらえ、通信時間の評価関数を設定し、通信の最小化を図ることとした。
【００２４】通信時間の最小化のアルゴリズムとして、焼き鈍し法（アニーリング）を採用することで並列計算機上の通信時間が最小となるようにするとよい。代表的なデータ参照関係の場合は、この最適化問題をユーザが解くことなく、最小な時間で、最適なマッピング情報を得られるようにするとよい。
【００２５】代表的なデータ参照関係とは、最近接参照、近接参照、全対全、等これまで経験的にアプリケーションプログラムに見られたものである。ユーザ固有のデータ参照関係の場合は、予め最適化問題を解き、最適マッピング情報データベースにこれを登録することで、代表的なデータ参照関係の場合と同様に、アプリケーションプログラムの中からデータベースにアクセスするだけで最適なマッピングを得られるようにするとよい。
【００２６】ユーザ固有のデータ参照関係とは、ユーザにきわめて強く依存したもので、前記したデータ参照パターン以外の汎用性にとぼしいものである。例えば、”２つ飛び”、とか、”ある領域のみ”とかいった極めてまれなデータ参照パターンである。
【００２７】最適マッピング情報データベースシステムの作成にあたっては、データの参照関係、計算ユニット単位のプロセッサ間データ量、プロセッサ台数を定義し、マッピングの最適化手法により、マッピング情報をデータベース出力する。
【００２８】データベースが構築されると、その後は、並列処理に当たって、このデータベースを参照し、計算モデルにふさわしいマッピングを選択して、処理を遂行する。
【００２９】
【作用】本発明では、前記図１のような、ユーザ・インターフェースを提供し、ユーザが分割したＮ次元モデルの計算ユニットと、物理プロセッサに識別符号をつけておき、その対応をアドレス変換テーブルでもつので、これにユーザがアクセスするだけで、自由にマッピング操作をすることが可能になる。
【００３０】その後は、プロセッサ間通信はユーザが分割ユニットの識別符号を用いて行うことが出来る。これで、ユーザ空間での作業が実現し、Ｎ次元計算モデルに応じてマッピングを自由に操作できるようになる。
【００３１】
【実施例】以下、本発明の好適具体例を図を参照して説明する。実施例では、並列計算機ＡＰ１０００を対象に行った研究について、このユーザ・インターフェース実現の方法を述べるとともに、典型的なデータ参照パターンについての、最適なマッピング方法について述べる。
＜ＡＰ１０００について＞まず、並列計算機ＡＰ１０００を、図２、図３に従って説明する。
【００３２】ＡＰ１０００は、各プロセッサが、２次元トーラス状に接続されているＭＩＭＤ（Multi Instruction stream Multi Data stream）型の並列コンピュータである。特徴として、３つの通信ネットワークを装備していて、プロセッサ（ＰＥ）間の通信はトーラスネットワークを用いる。ＰＥ間通信の最適化にあったては、ネットワークの特性を考慮しなければならない。
【００３３】図２にＡＰ１０００のアーキテクチャ構成図を示す。ＡＰ１０００は、１対１の通信に使用するトーラス・ネットワーク３１（Torus network）（以下、T-net という）、１対多の通信に使用する放送ネットワーク３３（Broadcast network）（以下、B-net という）、バリア同期専用の同期ネットワーク３５（Synchronization network）（以下、S-net という）の３種の独立した通信ネットワークをもつ。
【００３４】T-net はその２次元格子点にルーチング・コントローラ３７（ＲＴＣ）を有し、B-net は複数の放送ネットワーク・インターフェース３９（ＢＩＦ）を有している。T-net のＲＴＣ３７とB-net のＢＩＦ３９とは４対１の関係でそれぞれバスで接続され、各バス上に、それぞれセル・プロセッサ４１（以下、単にセルという）が設けられている。その内にいくつかには、オプション構成として、フレーム・バッファ４３と、ハード・ディスク４５とが接続されている。また、フレーム・バッファ４３にモニタ４７が接続されている。
【００３５】前記B-net の各ＢＩＦ３９は、S-net３５に一括して接続されている。また、ＢＩＦの一つにホストコンピュータ４９が接続されている。この結果、全セル４１とホスト４９は、B-netによって接続される。
【００３６】B-netは、階層バスとリングを組み合わせたネットワークで、データ放送、分散、収集に使用する。B-netは、データ転送中は一つのセルまたはホストに占有されるので、データ転送を行いたいホストまたはセルは、データ転送を行う前にB-netの使用要求を出し

使用権を獲得する。B-netは、３２－ｂｉｔのデータパスとリセット、割込みなどの制御信号から構成されており、パイプライン化されたハンドシェーク制御によって、５０ＭＢｙｔｅ／ｓのデータ転送レートをもつ。

【００３７】T-netは、２次元トーラス状のトポロジーをもつネットワークでメッセージの中継は、ワームホールルーチングによりハードウェアで自動的に行われる。ワームホールルーチングでは、メッセージのヘッダが、入力チャネルから出力チャネルへ中継ルートを作りながらメッセージを送り出す。ストアアンドフォワードルーチングが、中継プロセッサがメッセージ全体をストアするのに対し、ワームホールルーチングではフリットと呼ばれる数バイト（ビット、ＡＰ１０００では１６ビット）のデータのみが中継プロセッサにストアされるため、低レイテンシが実現できる。

【００３８】このように、ワームホールルーチングは、通信の遅延時間（レイテンシ：ltency）が小さいという特長をもつが、メッセージが通信チャネルをブロックするためデッドロック発生の可能性とスループットの低下という問題がある。ＡＰ１０００のT-netでは、ワームホールルーチングに構造化バッファプールアルゴリズムを組み合わせることによりデッドロックを回避しスループットの低下を抑えた。また、放送通信のレイテンシを小さくするため、行または列への放送機能も同時にインプリメントした。T-netのそれぞれのポートは、１６ｂｉｔ幅のデータパスをもち、B-netと同じようにパイプライン化されたハンドシェーク制御により、２５ＭＢｙｔｅ／ｓのデータ転送レートをもつ。

【００３９】全セルとホストは、S-netによっても接続されている。S-netは、ツリー状のトポロジーをもち、バリア同期とステータスの検出に使用される。各セルからのデータは、S-netの根の部分に向かって送られる。S-netの各ノードでは、各セルからのデータがＡＮＤ演算によってマージされる。全セルの出力したデータの理論積がS-netの根の部分で得られる。得られた結果は、今度はS-netを逆に進んで全セルに知らされる。各セルがS-netに送り出したデータの結果が得られるまでの時間は、８クロック（６４０ｎｓ）とセル数に関係なく一定である。

【００４０】ホストコンピュータには、汎用ワークステーションを使用している。ＡＰ１０００のホストインターフェースは、ホストコンピュータに実装するＶＭＥバスインターフェースボードとＡＰ１０００フレーム内に実装されるホストインターフェースボードから構成される。ホストインターフェースボードには、B-netインターフェースと３２ＭＢｙｔｅのローカルメモリが実装され、メッセージバッファとして利用できる。

【００４１】図３に前記セルの構成を示す。個々のセルは、整数演算ユニット５１（ＩＵ）、不動小数点演算ユニット５３（ＦＰＵ）、メッセージコントローラ５５（ＭＳＣ）、ルーチングコントローラ５７（ＲＴＣ）、B-netインターフェース５９（ＢＩＦ）、１６ＭＢｙｔｅのメインメモリ（ＤＲＡＭ）６１から構成される。ＩＵとＦＰＵは、１２８ＫＢｙｔｅのダイレクトマップキャシュメモリ６３に接続され、２５ＭＨｚで動作する。ＩＵには、ＳＰＡＲＣアーキテクチャのものを採用した。ＲＴＣには、２次元トーラスネットワーク上での自動ルーチング機能を、ＢＩＦには、データ分散収集及びバリア同期機能をインプリメントした。ＭＳＣ，ＲＴＣ，ＢＩＦ，とＤＲＡＭコントローラ６５（ＤＲＡＭＣ）は，ＬＢＵＳとよばれる３２ｂｉｔの内部バスで接続される。個々のセルのＬＢＵＳは、コネクタを介して外部に取り出されており、高速Ｉ／Ｏインターフェース、拡張メモリ、ディスクインターフェース、ベクタープロセッサ等の個々のオプション・ハードウェアが接続できるようになっている。

【００４２】メインメモリは4重インタリーブでＤＲＡＭを制御するＤＲＡＭＣと、４０個の４Ｍ－ＤＲＡＭで構成される。ＭＳＣはキャッシュコントローラ、ラインセンド（Line sending）とバッファレシーブと呼ばれる１対のメッセージハンドラー、４チャンネルの高機能ＤＭＡコントローラから構成される。

【００４３】本実施例では、このような構成のアーキテクチャ上に、図４のような、前記Ｎ次元計算モデルを識別符号を有する複数の計算ユニットに分割する計算モデル分割部７１と、前記複数の各プロセッサに識別符号をつけておき分割した計算ユニットと各プロセッサとの対応関係をアドレス変換テーブル上に形成する管理部７３と、分割した計算ユニットを、各プロセッサに任意にマッピングするマッピング部７５と、が実現される。そして、マッピング部７５で得られた最適マッピング情報はデータベース部７７に出力される。このデータベース部７７は、図２のハードディスク４５上に設けられる。以後は、このデータベース部７７を管理部７３を介して参照し、アドレス変換テーブルを構成して、並列計算を行う。

【００４４】ところで、ユーザ空間の重要性、また、そのユーザ空間からの自由なユーザ・マッピングの必要性は前記したところであるが、このような要請に応えるインターフェースをＡＰ１０００上に実現した。

【００４５】実現方法の概要は、図４で示したように、最初に、ユーザが分割したＮ次元モデルの計算ユニットと、物理プロセッサに識別符号をつけておき、その対応をアドレス変換テーブルでもつ方法である。すると、これにユーザがアクセスするだけで、自由にマッピング操作をすることが可能になる。その後は、プロセッサ間通信はユーザが分割ユニットの識別符号を用いて行うことが出来る。これで、ユーザ空間での作業が実現し、Ｎ次元計算モデルに応じてマッピングを自由に操作できるようになる。これを、従来のライブラリ群に加えて、図５

で示したようなＮ次元サブルーチン・ライブラリとして拡張、実現した。

【００４６】より詳細には、使用するプロセッサ数を宣言するステップ、宣言したプロセッサの数の範囲内の数＝$N^n$ に計算モデルを計算ユニットへと分割するステップ、分割した計算ユニットに識別符号を付与するステップ、使用するプロセッサに識別符号を付与するステップ、計算ユニットに付与された識別符号と、プロセッサに付与された識別符号とを対応づけて管理テーブルに登録するステップとによりユーザ・インターフェースが構築される。なお、分割された計算ユニットと、プロセッサに付与される識別符号は、１、２、３・・・というような通し番号が管理上好ましい。

【００４７】＜処理の流れ＞最適マッピングをするためには、図６のように、まず、計算モデルについての情報を入力する必要がある。すなわち、データの参照関係（データ参照パターンともいう）、計算ユニット単位のプロセッサ間データ量、プロセッサ台数を定義する（ステップＡ１）。

【００４８】そして、マッピングの最適化手法により、最適マッピングを得る（ステップＡ２）。マッピングの最適化手法については後記する。得られたマッピング情報をデータベース出力する（ステップＡ３）。

【００４９】＜アプリケーションプログラムでの使用例＞データベースが構築されると、その後は、アプリケーションプログラムなどにおいて計算モデルの並列処理に当たって、以下のような処理が行われる。まず、図７で示したように、ユニット分割次元数とマッピングテーブルの入力が行われる（ステップＢ１）。ここで、マッピングテーブルの入力とは、前記したユーザ・インターフェースの構築である。次に、最適マッピング情報データベースにアクセスして、計算すべき計算モデルにふさわしい最適マッピング情報を読み出し、システム環境を設定する（ステップＢ２）。ここでは、図４のアドレス変換テーブル２５上にマッピング・テーブルが構築される。

【００５０】この処理が終了すると、ユニット分割次元上での通信を行いつつ、並列的に処理が行われる。

＜計算モデルに対応したマッピング＞では、最適なマッピングを得る方法について説明する。

【００５１】マッピングは、図８に示したように、データを任意のＮ次元に分割したあと、これを最適にプロセッサに割り当てることである。図９は、２次元に分割すべきか、３次元に分割すべきか、マッピングとしてパターン１がよいのかパターン２がよいのかという検討すべきことを示している。

【００５２】マッピングに当たって考慮すべきことは、データ参照パターン、計算モデルの次元数、通信コストの評価関数である。

「データ参照パターン」データ参照パターンと通信と計算のバランスを考慮して、分割の仕方が決定される。

【００５３】マッピングの前提としてデータ分割が問題となるが、考慮すべきことは、分割によって分割ユニット間での通信がどうなるかということと、ユニット内の計算量を他ユニットとの通信量とのバランスである。

【００５４】例えば、あるLattice sizeの立体格子モデルがあり、これを分割して各セル（プロセッサ）に割り当てるとき、データ参照パターンが最近接（±Ｘ方向、±Ｙ方向、±Ｚ方向の合わせて６方向を参照する）であるとしたら、ＡＰ１０００は２次元トーラス構造なので、マッピングの容易性を考えて図１０（Ａ）のように分割する。これを２次元分割という。ところが、このような分割では問題がある。すなわち、２次元分割では、図１１（Ａ）のように、通信方向が４方向となり、立体モデルで表面積を通信量、体積を計算量と見るとき、分割数が増えていくと、すなわち、セル（プロセッサ）台数を増やしていくと、計算量に対して通信量の占める割合が増してゆく。

【００５５】そこで、図１０（Ｂ）のような分割方法を考慮してみた。これを３次元分割という。３次元分割では、図１１（Ｂ）のように、通信方向が６方向となり、最近接のデータ参照パターンに対応できる。このため、通信量の占める割合が２次元分割に比較して少なくなる。しかし、マッピングの仕方が難しくなる。

【００５６】以上をまとめると、単位通信方向の通信量の占める割合を計算量に対して減らすために、分割の方法を考え、分割によって発生した通信パターンを考慮にいれマッピングの容易さを考慮するということである。

【００５７】ところで、データ参照パターンは、アプリケーションによってさまざまである。最初から全ての場合について考えるのは無理であろう。そこで、図９のように、アプリケーションによく見られるデータ参照パターンによって分類してみることにする。

最近接格子：自分のデータ更新に必要なデータを最も近い隣の格子点から参照する必要がある格子である。±Ｘ方向、±Ｙ方向、±Ｚ方向の合わせて６方向を参照する。

【００５８】応用例として、構造解析、熱伝動、流体解析（有限要素法、中心差分）、物性（イジングスピン）、ＭＤ（モデキュラーダイナミクス）などの計算に利用できる。

近接格子：自分のデータの更新に必要なデータを近くで隣合っている（最近接＋斜め方向の）格子点から参照する必要がある格子をいう。±Ｘ方向、±Ｙ方向、±Ｚ方向の合わせて６方向に加えて、±ＺＸ方向、±ＹＺ方向、±ＸＹ方向の合わせて６方向、合計１２方向を参照する。

【００５９】応用例として、ＱＣＤ（Quantum chromodynamics：量子色力学）などに利用できる。

Ｎ対Ｎ：データの更新に全ての格子点を参照する必要

があるもので、応用例として、希薄流体の粒子追跡などに利用できる。希薄流体の粒子を追跡するとき、どの領域に粒子が行くか分からないので、全ての分割領域を参照する必要がある。

完全独立：通信が無い場合である。この場合、計算が並列計算機の全てのプロセッサ（ＰＥ）で独立に行える場合である。

【００６０】以上の各場合と、分割すべき次数との関係を図９に示す。以上のデータ参照パターンの相違により本発明でどのような影響があるかを、前記３次元分割の場合を例にととって説明する。

【００６１】最近接のとき：３次元分割によって各ユニットには６方向の通信が発生する。これを２次元のセル（プロセッサ）へマッピングするのであるが、２次元セルには４方向の通信経路しか存在しない。ＡＰ１０００では、セル間の距離に通信時間は依存しないのが建て前であるが、メッセージの競合を避けるためには互いに通信すべきユニットはできるだけ近くに置きたい。そこで、４方向を隣に置けるが、残りの２方向をどのように処理するかが問題となる。

【００６２】近接（斜め方向も含む）のとき：この場合、ユニットが計算に必要とする方向が増えるのでその分、マッピングの仕方に影響がでる。

Ｎ対Ｎの場合：この場合は、データの更新に全ての計算単位の参照が必要な場合であるから、その点を考慮する必要がある。この場合、分割によって膨大な通信量が発生するのでマッピングには近接と異なる考え方を導入する必要がある。

「計算モデルの次元数とマッピング」データの参照は、問題が何次元であるかによっても考え方が異なってくる。例えば図１０、図１１の様に、データ参照が最近接方向で２次元の計算モデルを、ＡＰ１０００のような２次元トーラスにマッピングする場合は、そのままマッピングすればいいが、３次元の計算モデルの場合だと、途中で変換が必要になる。計算モデルの次元が増える程、各計算ユニットが通信する方向は増えてゆき、高次元の計算モデルになるにつれて、難しくなることが想像される。例えば、３次元の最近接格子の場合は通信方向が６方向で、４次元だと８方向である。これを２次元にマッピングするには、残りの方向に当たる部分をどこかに持ってゆかなければならない。

【００６３】実施例では、ＡＰ１０００（２次元トーラス）での３次元の最近接格子の通信の最適化問題を考えることとした。これは、アプリケーションで最も応用範囲の広いケースである。

「通信コストの評価関数」一般的に、並列計算機で通信コスト（時間）の削減を考える時に、通信回数、通信量、通信の混み方、プロセッサ（ＰＥ）間距離を考えねばならない。どれが、どのくらい通信コストに、効いてくるかの比率は、その並列計算機のアーキテクチャに大きく依存している。ＡＰ１０００の場合は、プロセッサ間通信コストは、図１２、図１３に示されるように、プロセッサ（ＰＥ）間の距離にはあまり依存せず、通信の混み方に大きく依存している。

【００６４】しかし、通信が混んでくると、むしろ通信距離ができるだけ少ない方が、通信の衝突も起こりにくく混み方が少なくなり、通信コストが最小に近い値が得られるだろうと仮定した。

【００６５】このことは、図１４のＡＰ１０００のネットワーク特性からも想定できる。図１４の実験は、全てのセルが同じ方向、同じ距離だけ離れたセルに一斉に送信した場合メッセージの発信から受信までどれくらいかかるか？という実験である（メッセージの交換ではない）。これは３２×１６セルなのでトーラスの効用で最大送信距離（Ｘ，Ｙ）は図から明かなように２４となる。この結果から、データが大きくなり、経路上でメッセージの引きずりが起こるとき距離に対してリニアに通信時間が上がるように見える。そこで、これを確認するために、評価関数として各プロセッサ間の距離の総和Ｌを導入した。マッピングは、この距離の総和Ｌが最小となるように行う。ＡＰ１０００について、距離の総和と通信速度との関係を調べたところ、図１５のように、距離の総和が小さいほど、通信速度が速いことがわかった。

「マッピング」実施例では、サンプルとして、$3^3$の物理体系を$8^2$の２次元ネットワーク構成の計算機に、$4^3$を$8^2$にマッピングする場合について考える。しかし、全ての組み合せはそれぞれ、$10^{48}$通り、$10^{90}$通り存在するため、これら全てを調べるのは無理である。そこで、経験的に規則的な並びを考える方法と、シミュレーションにより求める方法を行った。

＊経験的マッピング

３次元の計算ユニットの、６つの通信方法全てに対し均等な通信コストは考えにくいので、まず、ある方向を優先的に隣になるようにおき、残りの方向についてなるべく近く、規則的になるように置くことを考えた。

【００６６】ＸＹＺ各方向に対して、立体的に分割し、それをマッピングするモデルを考える。図１６は$3^3$を$8^2$にマッピングした場合であり、図１７は$4^3$を$8^2$にマッピングした場合である。図１６、図１７に於いて、Ａのマッピング方法（スキップ法という）は、ＺＸ平面で４つに切っておき、Ｙ方向の各計算ユニットが正方形で隣合うように並べる方法で、ＺＸ方向の通信がひとつ飛びに並ぶ恰好になる。しかし、Ｙ方向が隣合う為には、ＺＸ平面を４つに分割しなければならない。

【００６７】より具体的には、（ａ）３次元計算モデルにおいて、Ｘ方向、Ｙ方向、Ｚ方向の計算単位をＶＣＸ、ＶＣＹ、ＶＣＺと定義し、Ｘ方向、Ｙ方向、Ｚ方向の分割数をＶＸ、ＶＹ、ＶＺと定義する。同時に、マッピング対象となる２次元物理プロセッサにおいて、Ｘ方

向物理プロセッサ台数、Ｙ方向物理プロセッサ台数をそれぞれＰＣＸ、ＰＣＹと定義するとともに、Ｘ方向、Ｙ方向に分割した分割数をそれぞれＰＸ、ＰＹと定義して物理プロセッサをグループ化する。

【００６８】そして、（ｂ）３次元計算モデルをＶＸ＝ＰＣＸ／２、ＶＹ＝ＰＣＹ／２、ＶＺ＝０に分割し、この分割によって出来た直方体ユニットをメインユニットと呼び、各分割単位に通し番号（メインユニット番号）を付与し、さらにこのメインユニットをＺ軸方向に４つに分割し、この結果できたユニットをサブユニットと呼んで、各分割単位に通し番号（サブユニット番号）を付与し、同時に、２次元物理プロセッサにおいて、ＰＸ＝ＰＣＸ／２、ＰＹ＝ＰＣＹ／２に分割してプロセッサをグループ化し、各分割単位に通し番号（グループ番号）を付与する。

【００６９】さらに、（ｃ）３次元モデルのメインユニット番号と同一番号のグループ番号を有する物理プロセッサのグループに、図１８のように、サブユニット番号０から３を、０と１、１と２、２と３、３と０とが隣接するように分配してマッピングする。図１６、図１７に於いて、Ｂのマッピング方法（タイル法という）は、ＺＸ平面でスライスしたものを順番に置いていったものである。これで、ＺＸ方向が隣合いＹ方向が均等（スライス数分離れる）に並ぶ。

【００７０】より具体的には、（ａ）３次元計算モデルにおいて、Ｘ方向、Ｙ方向、Ｚ方向の計算単位をＶＣＸ、ＶＣＹ、ＶＣＺと定義し、Ｘ方向、Ｙ方向、Ｚ方向の分割数をＶＸ、ＶＹ、ＶＺと定義する。同時に、マッピング対象となる２次元物理プロセッサにおいて、Ｘ方向物理プロセッサ台数、Ｙ方向物理プロセッサ台数をそれぞれＰＣＸ、ＰＣＹと定義するとともに、Ｘ方向、Ｙ方向に分割した分割数をそれぞれＰＸ、ＰＹと定義して物理プロセッサをグループ化することを前提とする。

【００７１】そして、（ｂ）３次元計算モデルをＶＸ＝０、ＶＹ＝０、ＶＺ＝ＶＣＺに分割し、この分割によって出来た直方体ユニットをメインユニットと呼び、各分割単位に通し番号（メインユニット番号）を付与し、このメインユニットのＸ方向計算単位数、Ｙ方向計算単位数をｖｘ、ｖｙとする。同時に、２次元物理プロセッサにおいて、ＰＸ＝ＰＣＸ／ｖｘ、ＰＹ＝ＰＣＹ／ｖｙに分割してプロセッサをグループ化し、各分割単位に通し番号（グループ番号）を付与する。

【００７２】さらに、（ｃ）ｖｘ＝ＰＸ、ｖｙ＝ＰＹの位置に対応して、図１９のように、メインユニットを２次元物理プロセッサのＸ方向に通し番号順に配置し、そのライン上でのＸ方向の端部物理プロセッサに来たら、Ｙ方向へ移行し、今度はメインユニットを逆のＸ方向に通し番号順に配置し、再度Ｘ方向の端部物理プロセッサに来たら、Ｙ方向へ移行し、以後、以上の蛇行によるマッピングを繰り返す。

＊シミュレーションによるマッピング

次に、シミュレーションによって最適なマッピングを求めようという方法を用いた。つまり、今回のマッピングの問題を距離の総和を最小化する組合わせの最適化問題として考えた。

【００７３】マッピングによる通信コストは、すべての最近接間格子間の距離の総和にほかならないとした。方法は、初期状態としてランダムにマッピングしておき、経路の総和を求めておく。図２０のように、どこか一組を取り替えてみて、経路の総和が減ったら交換を採用する。交換を繰り返していき、総和の変化が少なくなった状態を近似解として採用する方法である。

【００７４】具体的には、隣接格子点ＰｉＰｊがそれぞれ２次元格子点上のある点（Ｘｉ，Ｙｉ）（Ｘｊ，Ｙｊ）にマッピングされたとする。ＰｉＰｊの距離ｌｉｊは

ｌｉｊ＝｜Ｘｉ－Ｘｊ｜＋｜Ｙｉ－Ｙｊ｜

よって評価関数は、Ｌ＝Σｌｉｊになりこれが最小になった時の、マッピング状態を最適なものと近似する。

（最小なのは全ての格子点が一つになったときであるが、交換で格子点が重なることは考えない）ところが、このような緩和法には、局所的最小の存在が確認されており、△Ｌが最小だからといって必ずしも、通信コストが最小であるとは限らない。

＊アニーリング法

そこで、組合わせの最適化の近似解法として、カークパトリックら（Ｓ．Ｋｉｒｋｐａｔｒｉｃｋ　ｅｔ　ａｌ．1983）によって提案された、シミュレーテドアニーリング法（ｓｉｍｕｌａｔｅｄ　ａｎｎｅａｌｉｎｇ　ｍｅｔｈｏｄ）［４］を用いた。これは、評価関数をより小さくするように変化させていく過程に、局所的最小を脱出するような確率を導入した方法である。その確率をω、目安をＴ（確率を決定するパラメータ）とする。

【００７５】

ω（△Ｌ）＝ｅｘｐ（－△Ｌ／Ｔ），△Ｌ＞０

ω（△Ｌ）＝１　　，△Ｌ＜０

△Ｌ＝Ｌ（交換前）－Ｌ（交換後）：距離の変化

Ｔ（確率を決定するパラメータ）を段々下げてゆく（確率を下げてゆく）と評価関数が最小に近くなる。Ｔの下げかた（アニーリング・スケジュール）は、ギーマン兄弟（Ｓ．Ｇｅｍａｎ　ａｎｄ　Ｄ．Ｇｅｍａｎ　1984）による、

Ｔ（ｔ）＝Ｂ／ｌｎ　ｔ

Ｔ（ｔ）→０　　（ｔ→無限大）とした。

【００７６】Ｂ：評価関数の障壁の高さ

図２１に、アニーリング法のアルゴリズムを示す。すなわち、初期マップ位置を決定した後（ステップ１０１）、マップ位置を変化させ（ステップ１０２）、長さの変化（前記距離の総和の変化）△Ｌを算出し（ステップ１０３）、△Ｌ＜０のとき、前記マップ位置の

変更を採用し（ステップ１０４、１０５）、△Ｌ＜０でないとき、乱数Ｘ（０＜Ｘ＜１）を引き出し（ステップ１０６）、ｅｘｐ（－△Ｌ／Ｔ）を演算し（Ｔは確立を決める目安となるパラメータであり、例えば自然数を代入できる）（ステップ１０７）、ｅｘｐ（－△Ｌ／Ｔ）＜Ｘのとき、マップ位置の変更を採用せず（ステップ１０８）、ｅｘｐ（－△Ｌ／Ｔ）＜Ｘでないとき、マップ位置の変更を採用することとし（ステップ１０５）、以上からの処理を、前記Ｔの値を減少させて（ステップ１０９）繰り返すことにより、最終的に最適なマッピングを得ることができる。からまでの繰返し回数は、経験則から、１００万回程度で行うと、Ｌが十分に下がることがわかった。よって、ループの繰返し回数を予め入力して最適なＬを求めることができる。なお、前記が一定回数連続して繰り返されたとき、前記処理ループを中断させるブレークポイントを設けてもよい。

【００７７】このアニーリング法でどのように通信時間が減るかを、図２２に示す。

＜結果＞このアニーリングによって得られた、マッピング結果、$3^3 \rightarrow 8^2$の場合を図２３、$4^3 \rightarrow 8^2$の場合を図２６に示す。

（図中の番号はＸＹＺ各座標を表す）同時に、前記した経験マッピングのＡパターン方法（スキップ法）、Ｂパターン方法（タイル法）についてもマッピング結果を得た。$3^3 \rightarrow 8^2$の場合をそれぞれ図２４、図２５、$4^3 \rightarrow 8^2$の場合をそれぞれ図２７、図２８に示す。

＜マッピングの評価＞図２９に示したように、シミュレーションで、得られたマッピングと経験的に得られたマッピングとを距離の総和（Ｌ）で比較してみた。

【００７８】$3^3 \rightarrow 8^2$の場合は、経験的に得られたマッピングより、本手法でのマッピングの方が距離の総和が少なく良い結果が得られた。ただし、最良なマッピングであるとは限らない。$4^3 \rightarrow 8^2$の場合は、経験的に得られたマッピングと同程度の結果が得られた。

＜データベースへの格納＞以上のマッピングで得られた最適マッピングに関する情報は、図３０のようなファイル形式でデータベースに登録される。

【００７９】このデータベースがどのように使用されるかは、前記したように、図７で示した通りである。

＜結論＞評価関数として距離をとり、３次元格子の最近接参照モデルを２次元のトーラス・ネットワークにマッピングする時、アニーリング法を用いる事で、結果を比較的容易に経験的マッピングより良い結果が得られる場合があることを示した。今後、評価関数を正確化し改良することで、経験的手法が働かないような、高次元の問題のマッピングを考えるのに有効と考えられる。

【００８０】なお、本実施例は、ＡＰ１０００について行ったが、本発明により前記ユーザ・インターフェースを実現できる並列計算機であれば、他の並列計算機でも本発明を実施できることはいうまでもない。

＜マッピング最適化の並列処理＞図３１に、以上説明したマッピングの最適化を並列処理で行った場合の例を示した。

【００８１】ここでは説明を簡単にするため、２台のプロセッサで並列処理を行ったものとする。まず、親プロセッサがマップ位置等の初期値を子プロセッサに放送する（ステップ１０１－１）。また、親プロセッサでは、障壁サンプルの設定を行い（ステップ２０６）、障壁Ｂを子プロセッサに送信する（ステップ２０７）。このステップ２０６、２０７を繰返し、最終フラグを子プロセッサに送信する（ステップ２０８）。最終フラグとは以下のループを行うか行わないかのフラグである。

【００８２】子プロセッサでは、初期値を受信して初期設定を行う（ステップ１０１－２）。ついで、前記障壁Ｂを受信し（ステップ２０１）、マップ位置を変化させる（ステップ１０２）。ついで、長さの変化（前記距離の総和の変化）△Ｌを算出する（ステップ１０３）。△Ｌ＜０のとき、前記マップ位置の変更を採用する（ステップ１０４、１０５）。△Ｌ＜０でないとき、乱数Ｘ（０＜Ｘ＜１）を引き出し（ステップ１０６）、ｅｘｐ（－△Ｌ／Ｔ）を演算する（Ｔは確立を決める目安となるパラメータで、例えば自然数を代入できる）（ステップ１０７）。ｅｘｐ（－△Ｌ／Ｔ）＜Ｘのとき、マップ位置の変更を採用しない（ステップ１０８）。ｅｘｐ（－△Ｌ／Ｔ）＜Ｘでないとき、マップ位置の変更を採用する（ステップ１０５）。その後、Ｔの値を減少させ（ステップ１０９）、減少率が十分低いか否かを判定し（ステップ２０３）、目標値に達していない場合、ステップ２０１から１０９までの処理を繰り返す。

【００８３】ステップ２０１で最終フラグを受信した場合は、ステップ１０２から２０２の処理を飛ばして、全サンプル中での最小値を計算する（ステップ２０３）。そして、全サンプル中で「ＰＥ間距離の総和」が最小であるか否かを判定し（ステップ２０４）、全プロセッサの中で最小であればその結果を親プロセッサに送信する（ステップ２０５）。親プロセッサでは、最小マッピングの出力をして（ステップ２１０）、処理を終了する。なお、子プロセッサのステップ２０４で全サンプル中で最小でないとされた場合、結果は親プロセッサに送信せず、処理を終了する。ここで、全サンプルとは、全セルでのそれぞれのサンプル、すなわち、あるセルでの評価関数が最小となったマッピングパターンである。

【００８４】この並列化処理により、マッピングの最適化が高速で行われる。

【００８５】

【発明の効果】本発明では、ユーザ空間において、並列計算機のアーキテクチャを意識することなく最適なマッピングをすることができる。よって、並列処置の高速化を簡単に実現できる。

【図面の簡単な説明】
【図１】　本発明で構成されるユーザ・インターフェースを示す概念図
【図２】　実施例で使用した並列計算機のアーキテクチャを示す構成図
【図３】　実施例のセル構成図を示した構成図
【図４】　インターフェースと本発明の機能ブロックとの関係を示した図
【図５】　ライブラリプログラムの一部を示した図
【図６】　本実施例における処理フロー図
【図７】　得られたデータベースの使用例を示す処理フロー図
【図８】　データ分割とマッピングパターン選択の考え方を示した図
【図９】　データ参照パターンと分割次元数との関係を示す図
【図１０】　データ分割・マッピングの関係の一例を示した図
【図１１】　データ分割・マッピングの関係の一例を示した図
【図１２】　ＡＰ１０００におけるＰＥ間距離対通信時間を示す図
【図１３】　ＡＰ１０００における通信コンテンション対通信時間を示す図
【図１４】　ＡＰ１０００における距離と時間との関係を示す図
【図１５】　ＡＰ１０００における距離の総和と通信時間との関係を示す図
【図１６】　$3^3$を$8^2$にマッピングした場合を示す図
【図１７】　$4^3$を$8^2$にマッピングした場合を示す図
【図１８】　マッピング例を示す図
【図１９】　マッピング例を示す図
【図２０】　最適マッピングの検索例を示す図
【図２１】　シュミレーテド・アニーリング法のアルゴリズムを示す図
【図２２】　評価関数（距離）による通信時間の差を示す図
【図２３】　$3^3$を$8^2$にアニーリング法でマッピングした場合の具体的結果を示す図
【図２４】　$3^3$を$8^2$にスキップ法でマッピングした場合の具体的結果を示す図
【図２５】　$3^3$を$8^2$にタイル法でマッピングした場合の具体的結果を示す図
【図２６】　$4^3$を$8^2$にアニーリング法でマッピングした場合の具体的結果を示す図
【図２７】　$4^3$を$8^2$にスキップ法でマッピングした場合の具体的結果を示す図
【図２８】　$4^3$を$8^2$にタイル法でマッピングした場合の具体的結果を示す図
【図２９】　マッピングの手法による評価関数と実測の通信時間との関係を示した図
【図３０】　最適マッピング情報データベースのファイル形式を示した図
【図３１】　マッピングの最適化を並列処理で行った場合のフローチャート図
【符号の説明】
２１・・・Ｎ次元計算モデル
２３ＰＥ・・・物理プロセッサ
２５・・・アドレス変換テーブル
３１・・・トーラス・ネットワーク
３３・・・放送ネットワーク
３５・・・同期ネットワーク
３７・・・ルーチング・コントローラ
３９・・・放送ネットワーク・インターフェース
４１・・・セル・プロセッサ
４１・・・全セル
４３・・・フレーム・バッファ
４５・・・ハード・ディスク
４７・・・モニタ
４９・・・ホストコンピュータ
５１・・・整数演算ユニット
５３・・・不動小数点演算ユニット
５５・・・メッセージコントローラ
５７・・・ルーチングコントローラ
５９・・・インターフェース
６３・・・ダイレクトマップキャシュメモリ
６５・・・ＤＲＡＭコントローラ
７１・・・計算モデル分割部
７３・・・管理部
７５・・・マッピング部
Ｌ・・・距離の総和

【図１】

【図２】

【図３】

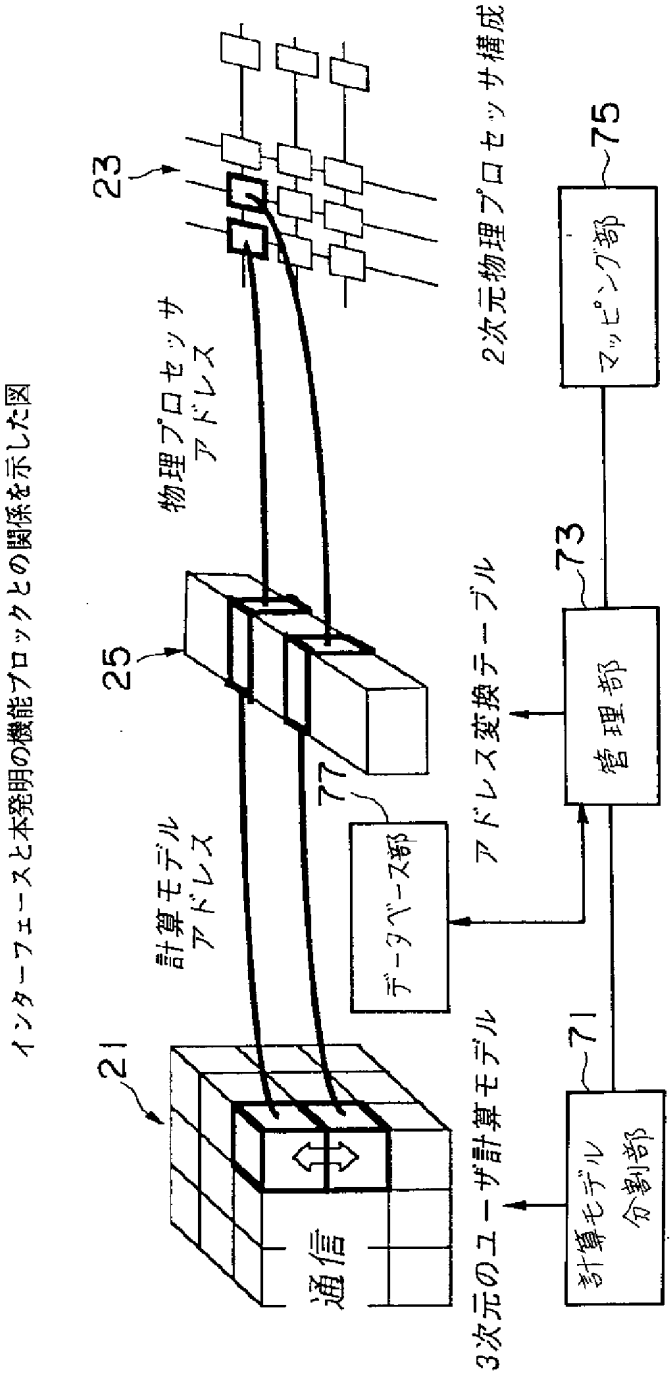
【図５】

ライブラリプログラムの一部を示した図

*N次元通信ライブラリ使用例*

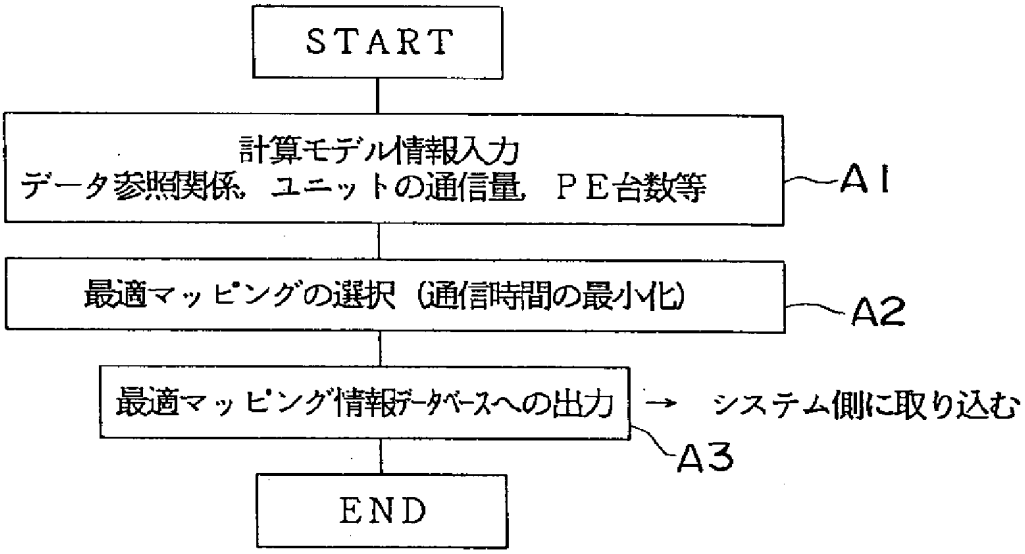
```
CALL  CONF(3次元,      ·····)  ──→  3次元ユーザ空間を定義

CALL  SEND(2,0,0,      ·····)
CALL  RECV(2,0,1,      ·····)  ──→  3次元ユーザ空間上で
                                    データの送受信を行う
```

【図4】

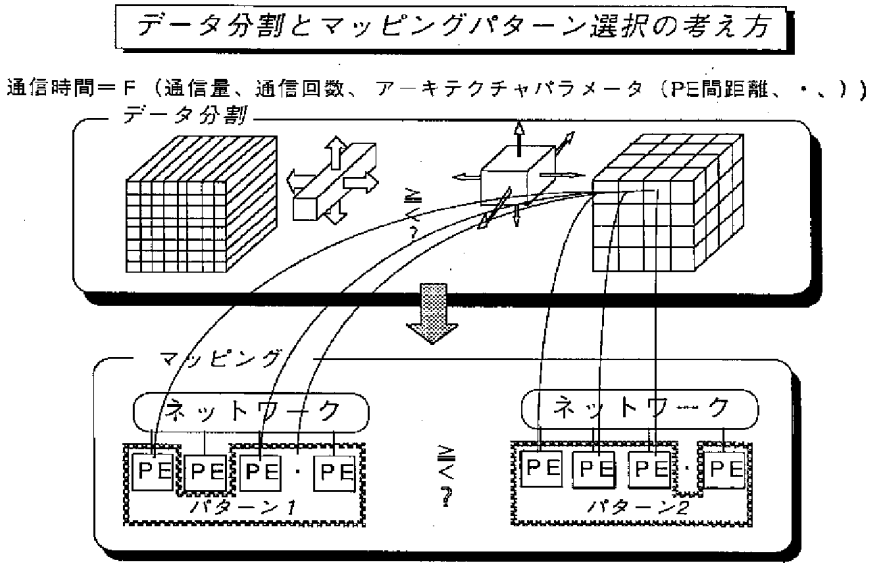

インターフェースと本発明の機能ブロックとの関係を示した図
2次元物理プロセッサ構成
3次元のユーザ計算モデル
アドレス変換テーブル
物理プロセッサ
アドレス
計算モデル
アドレス
通信
データベース部
管理部
マッピング部
計算モデル
分割部
21
23
25
71
73
75
77

【図6】

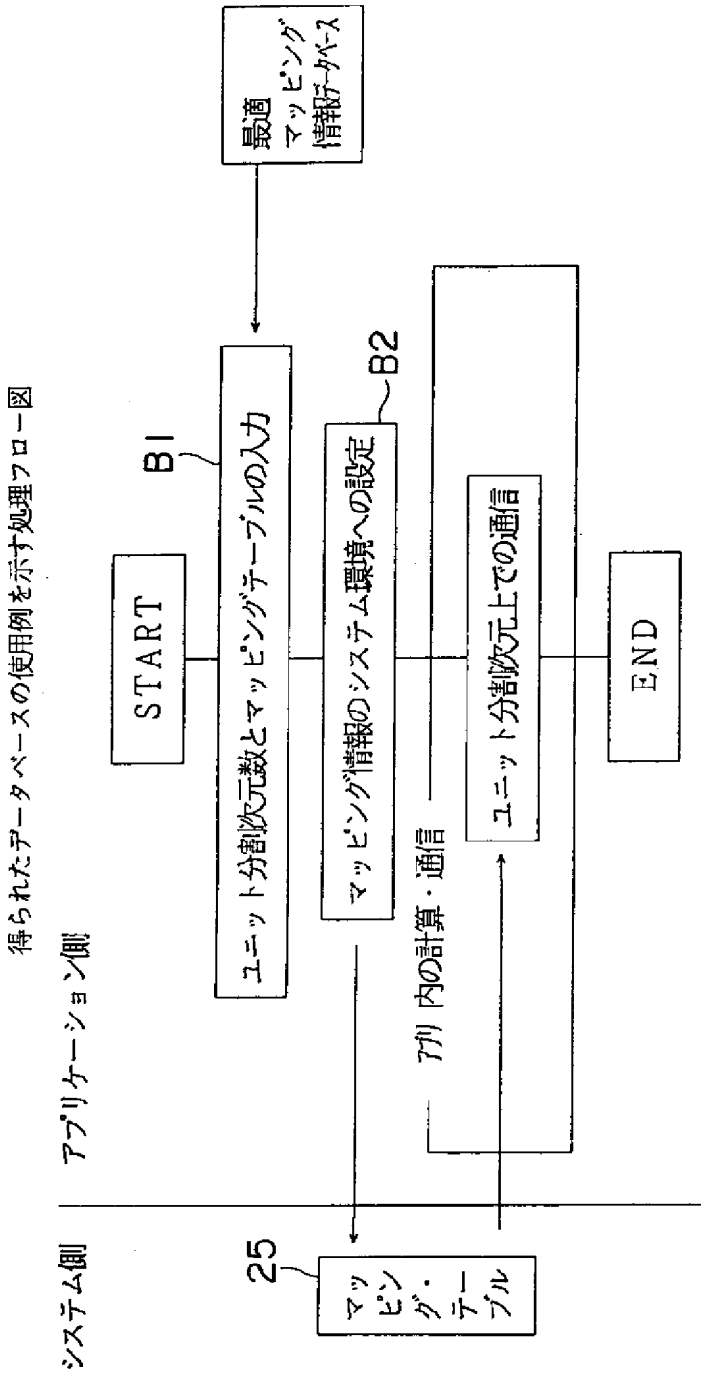
本実施例における処理フロー図



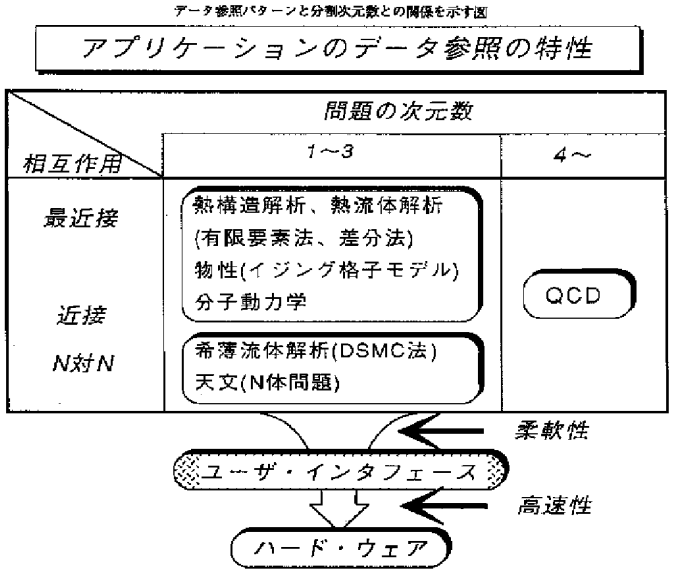

【図8】



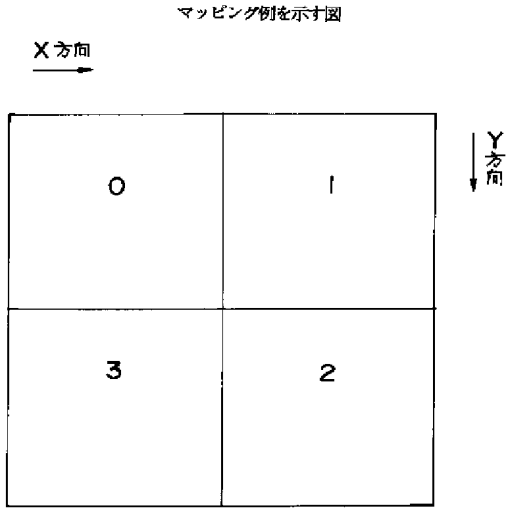
【図7】


得られたデータベースの使用例を示す処理フロー図
アプリケーション側
システム側
START
B1
ユニット分割次元数とマッピングテーブルの入力
最適
マッピング
情報データベース
B2
マッピング情報のシステム環境への設定
プリ 内の計算・通信
ユニット分割次元上での通信
END
25
マッピング・テーブル

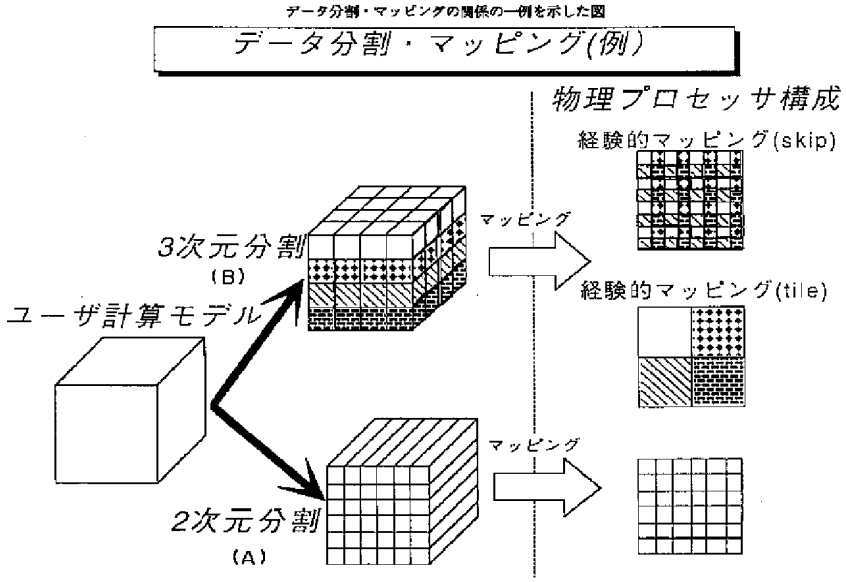
【図9】


データ参照パターンと分割次元数との関係を示す図
アプリケーションのデータ参照の特性
問題の次元数
相互作用
1～3
4～
最近接
近接
N対N
熱構造解析、熱流体解析
(有限要素法、差分法)
物性(イジング格子モデル)
分子動力学
QCD
希薄流体解析(DSMC法)
天文(N体問題)
柔軟性
ユーザ・インタフェース
高速性
ハード・ウェア

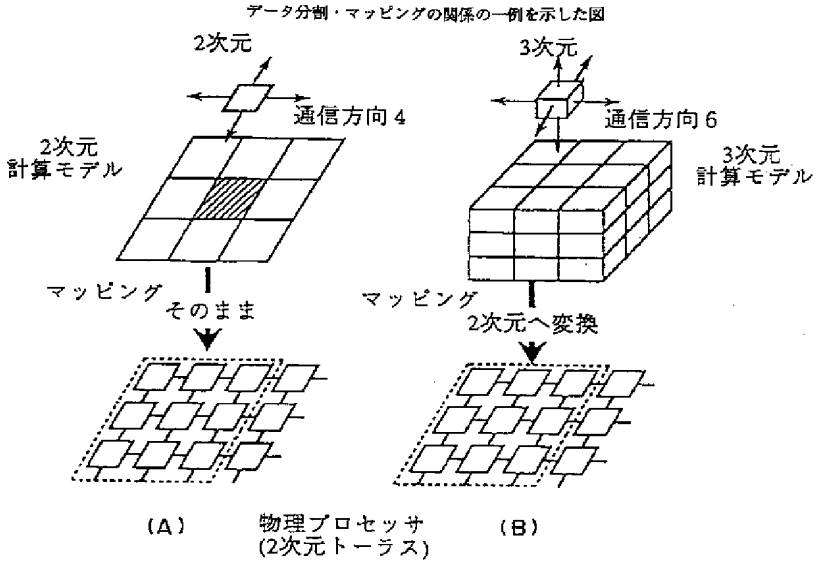
【図18】


マッピング例を示す図
X方向
Y方向
0
1
3
2

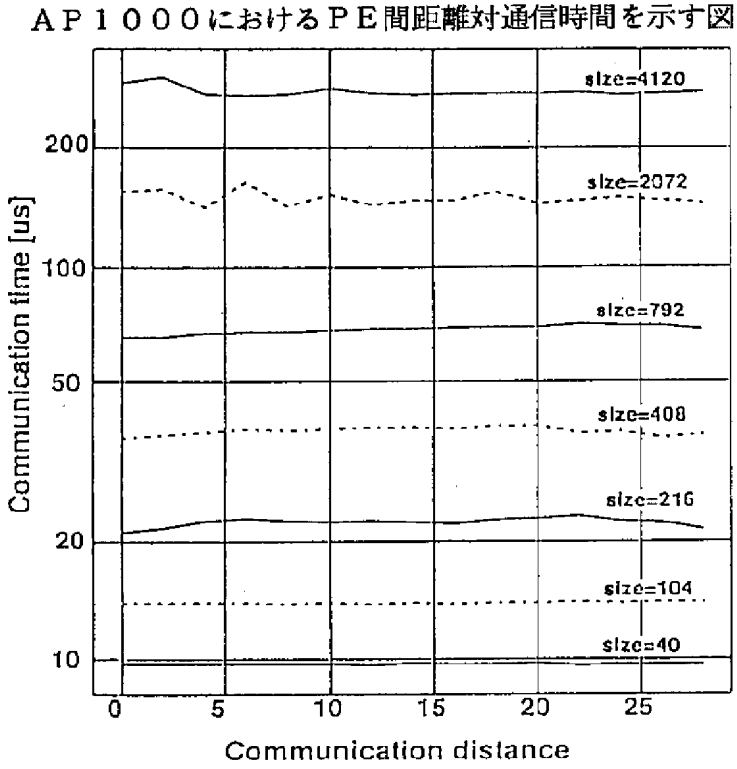
【図10】


データ分割・マッピングの関係の一例を示した図
データ分割・マッピング(例)
物理プロセッサ構成
経験的マッピング(skip)
マッピング
3次元分割
(B)
経験的マッピング(tile)
ユーザ計算モデル
マッピング
2次元分割
(A)

【図１１】

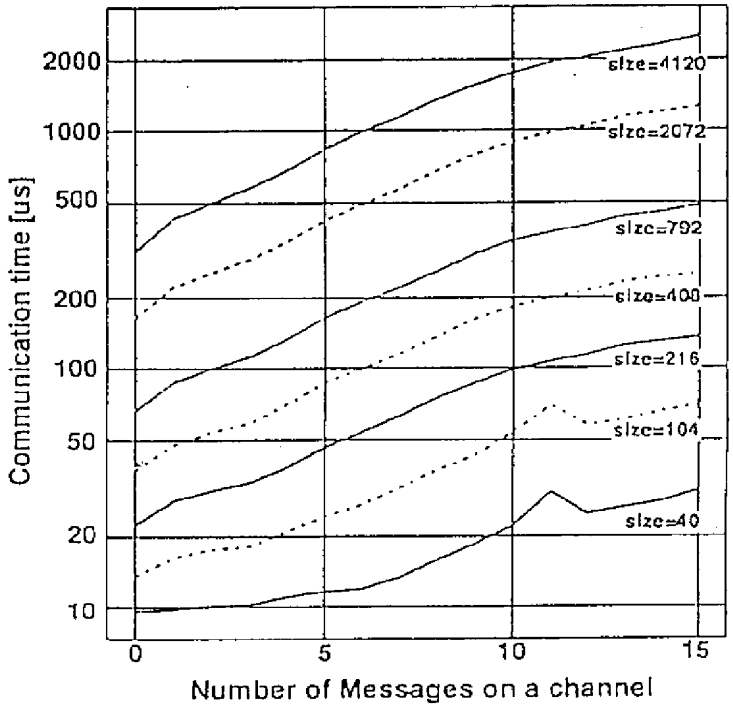

データ分割・マッピングの関係の一例を示した図
2次元
通信方向４
2次元
計算モデル
マッピング
そのまま
3次元
通信方向６
3次元
計算モデル
マッピング
2次元へ変換
(A)
物理プロセッサ
(2次元トーラス)
(B)

【図１２】

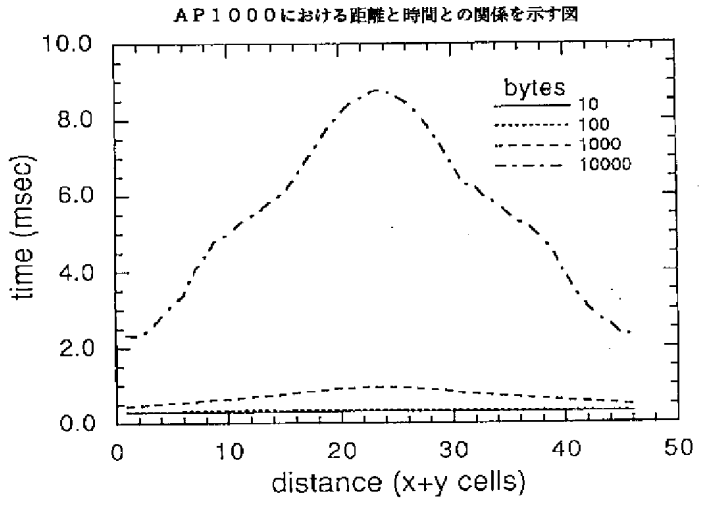

ＡＰ１０００におけるＰＥ間距離対通信時間を示す図
Communication time [us]
200
100
50
20
10
size=4120
size=2072
size=792
size=408
size=216
size=104
size=40
0
5
10
15
20
25
Communication distance

【図１３】

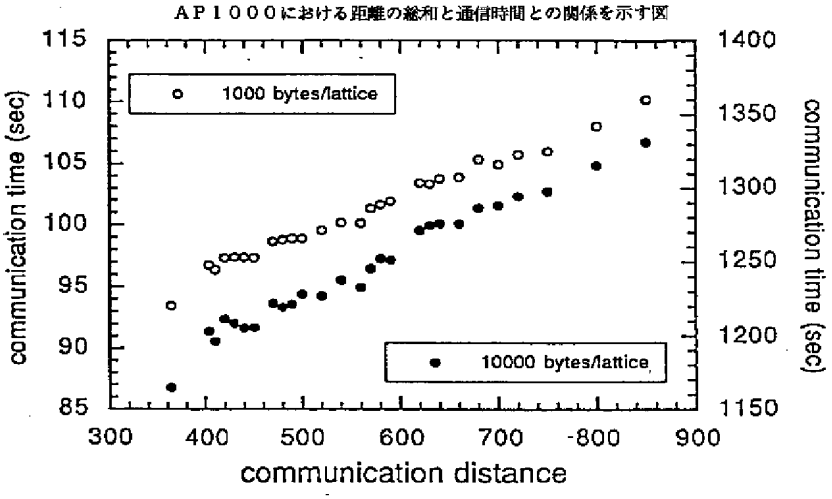
ＡＰ１０００における通信コンテンション対通信時間を示す図



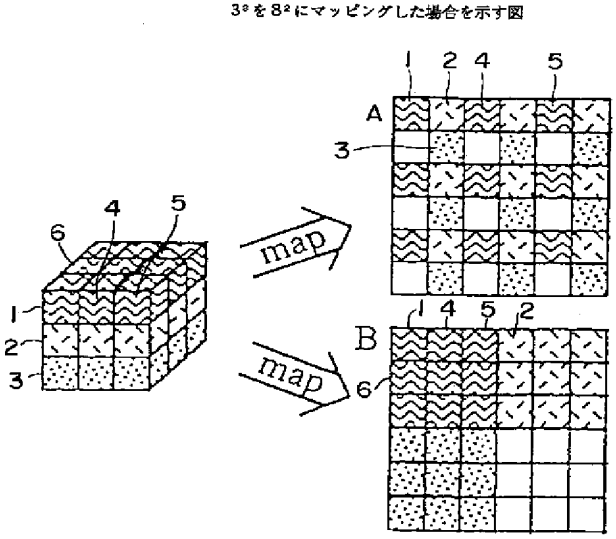

【図１４】



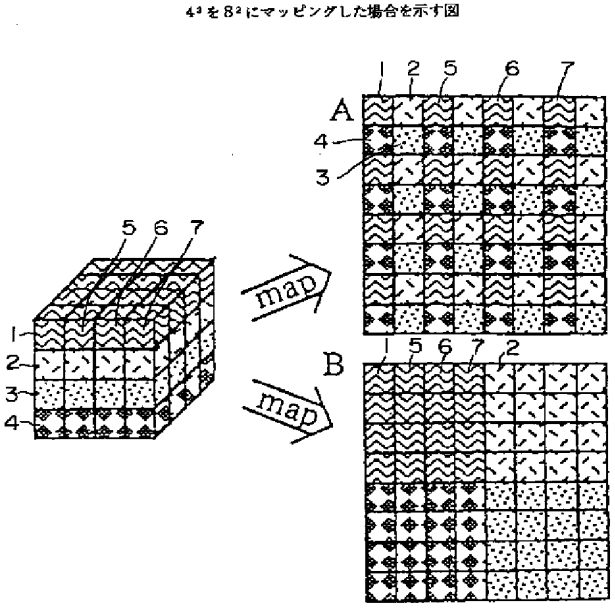
【図１５】


APｌ０００における距離の総和と通信時間との関係を示す図
115
110
105
100
95
90
85
1400
1350
1300
1250
1200
1150
1000 bytes/lattice
10000 bytes/lattice
communication time (sec)
communication time (sec)
300
400
500
600
700
800
900
communication distance

【図１６】

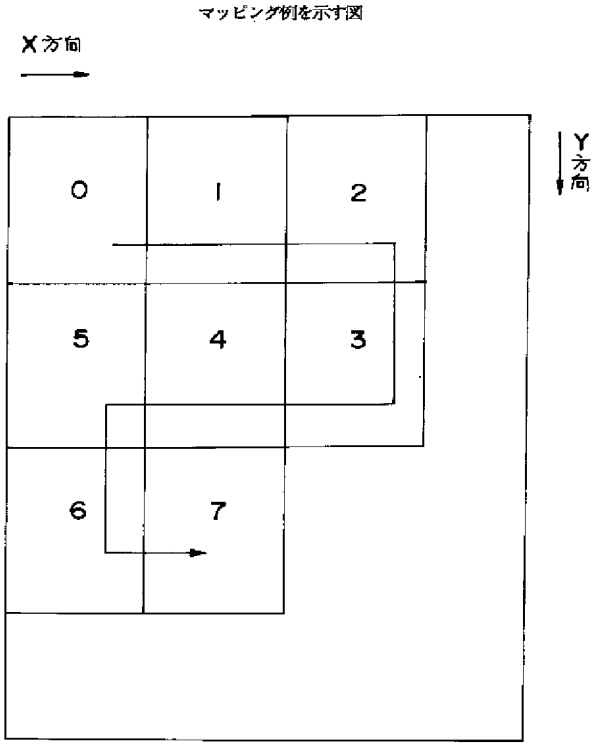

3³を8²にマッピングした場合を示す図
map
map
A
B

【図１７】


4³を8²にマッピングした場合を示す図
map
map
A
B

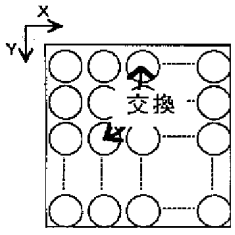
【図１９】

マッピング例を示す図




【図２０】

最適マッピングの検索例を示す図




マッピングパターン

最適化手法:シミュレーティド・アニーリング法
評価関数　:距離(L)

$$L = \Sigma \{ | X(i)-X(j) | + | Y(i)-Y(j) | \}$$

ΔL＝L(交換前)－L(交換後)

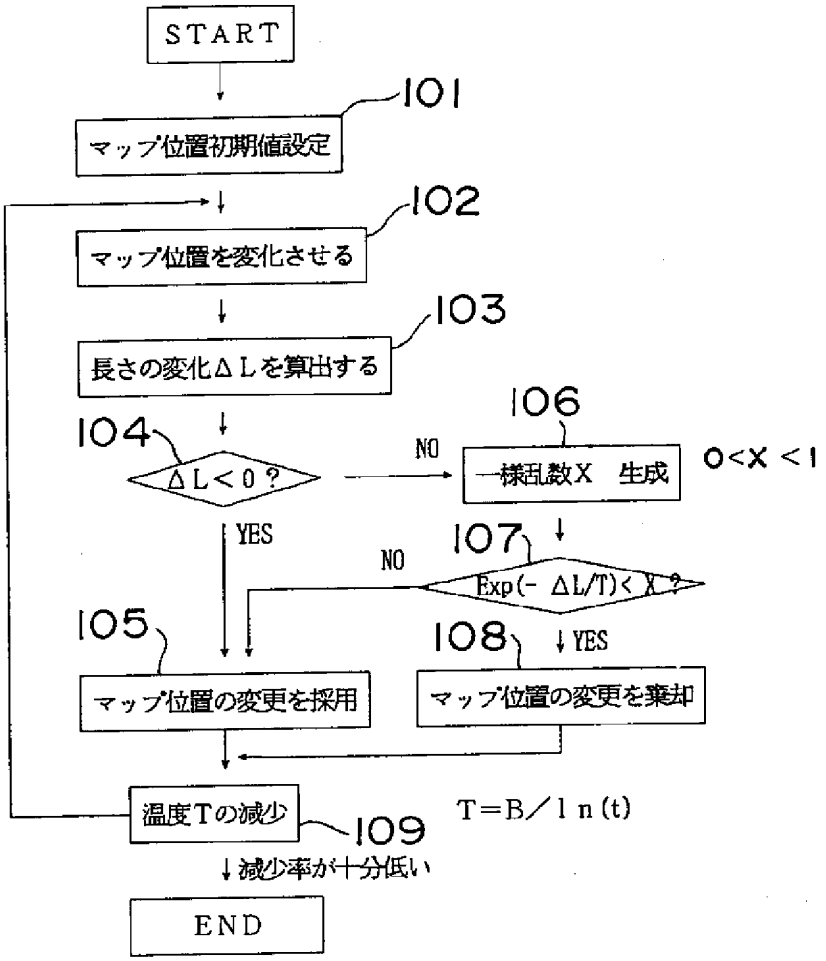
$$\omega(\Delta L) = \exp(-\Delta L / T) \quad : \Delta L > 0$$
$$\omega(\Delta L) = 1 \quad : \Delta L \leqq 0$$

ω：確率、Ｔ＝温度(確率ωの目安)

アニーリング・スケジュール　Ｔ＝Ｂ／ln t
Ｂ：評価関数の障壁の高さ

【図２１】

シュミレーテド・アニーリング法のアルゴリズムを示す図

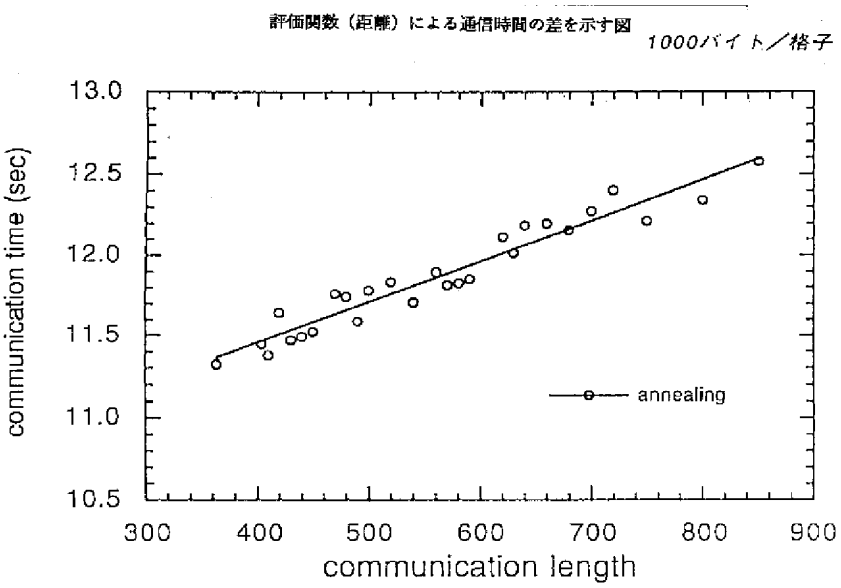



【図２９】

マッピングの手法による評価関数と実測の通信時間との関係を 示した図

| 手法 | マッピング パターン | 系 | 距離の総和Ｌ |
|---|---|---|---|
| 頭で考えた マッピング | A | $3^3 \rightarrow 8^2$ | 180 |
| | | $4^3 \rightarrow 8^2$ | 320 |
| | B | $3^3 \rightarrow 8^2$ | 180 |
| | | $4^3 \rightarrow 8^2$ | 448 |
| アニーリング | C | $3^3 \rightarrow 8^2$ | 154 |
| | D | $4^3 \rightarrow 8^2$ | 320 |

【図22】




【図23】

$3^3$ を $8^2$ にアニーリング法でマッピングした場合の具体的結果を示す図

```
1  mapfile is a27.154
2  # 3 dimensional mapping
3  # 10000 bytes per a lattice
4  # 100 steps
5  # send-direction 6
6  # LOCELX= 3
7  # LOCELY= 3
8  # LOCELZ= 3
9  (-,-,-) (-,-,-) (-,-,-) (-,-,-) (-,-,-) (-,-,-) (-,-,-) (-,-,-)
10 (-,-,-) (-,-,-) (-,-,-) (-,-,-) (-,-,-) (-,-,-) (-,-,-) (-,-,-)
11 (-,-,-) (-,-,-) (-,-,-) (-,-,-) (-,-,-) (-,-,-) (-,-,-) (-,-,-)
12 (-,-,-) (-,-,-) (-,-,-) (2,2,1) (1,2,1) (-,-,-) (-,-,-) (-,-,-)
13 (0,2,0) (2,2,0) (1,2,0) (2,1,1) (2,2,2) (1,2,2) (0,2,2) (0,2,1)
14 (0,1,0) (2,1,0) (1,1,0) (1,1,1) (2,1,2) (1,1,2) (0,1,2) (0,1,1)
15 (0,0,0) (2,0,0) (1,0,0) (1,0,1) (2,0,2) (1,0,2) (0,0,2) (0,0,1)
16 (-,-,-) (-,-,-) (-,-,-) (2,0,1) (-,-,-) (-,-,-) (-,-,-) (-,-,-)
17 # creating task on 27 PE
18 # sending parameter
19 # sending logical position to each cell
20 # sending logical mapping
21 # waiting stable
22 *** HOST_EXIT(0) ***
23 -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
24 Max: 2.037274 Min: 2.033902 Ave: 2.035016
```

マッピングパターン

実測時間

【図24】

$3^3$ を $8^2$ にスキップ法でマッピングした場合の具体的結果を示す図

```
26  mapfile is map.skip.27
27  # 3 dimensional mapping
28  # 10000 bytes per a lattice
29  # 100 steps
30  # send-direction 6
31  # LOCELX= 3
32  # LOCELY= 3
33  # LOCELZ= 3
34  (0,0,0) (0,0,1) (0,1,0) (0,1,1) (0,2,0) (0,2,1) (-,-,-) (-,-,-)
35  (-,-,-) (0,0,2) (-,-,-) (0,1,2) (-,-,-) (0,2,2) (-,-,-) (-,-,-)
36  (1,0,0) (1,0,1) (1,1,0) (1,1,1) (1,2,0) (1,2,1) (-,-,-) (-,-,-)
37  (-,-,-) (1,0,2) (-,-,-) (1,1,2) (-,-,-) (1,2,2) (-,-,-) (-,-,-)
38  (2,0,0) (2,0,1) (2,1,0) (2,1,1) (2,2,0) (2,2,1) (-,-,-) (-,-,-)
39  (-,-,-) (2,0,2) (-,-,-) (2,1,2) (-,-,-) (2,2,2) (-,-,-) (-,-,-)
40  (-,-,-) (-,-,-) (-,-,-) (-,-,-) (-,-,-) (-,-,-) (-,-,-) (-,-,-)
41  (-,-,-) (-,-,-) (-,-,-) (-,-,-) (-,-,-) (-,-,-) (-,-,-) (-,-,-)
42  # creating task on 27 PE
43  # sending parameter
44  # sending logical position to each cell
45  # sending logical mapping
46  # waiting stable
47  *** HOST_EXIT(0) ***
48  -=-=-=-
```

【図26】

$4^3$を$8^2$にアニーリング法でマッピングした場合の具体的結果を示す図

```
 1  mapfile is a64.320
 2  # 3 dimensional mapping
 3  # 10000 bytes per a lattice
 4  # 100 steps
 5  # send-direction 6
 6  # LOCELX= 4
 7  # LOCELY= 4
 8  # LOCELZ= 4
 9  (3,1,0) (2,2,0) (3,2,0) (3,2,3) (2,2,3) (3,1,3) (2,1,3) (2,1,0)
10  (3,1,1) (2,2,1) (3,2,1) (3,2,2) (2,2,2) (3,1,2) (2,1,2) (2,1,1)
11  (0,1,1) (1,2,1) (0,2,1) (0,2,2) (1,2,2) (0,1,2) (1,1,2) (1,1,1)
12  (3,0,1) (2,3,1) (3,3,1) (3,3,2) (2,3,2) (3,0,2) (2,0,2) (2,0,1)
13  (0,0,1) (1,3,1) (0,3,1) (0,3,2) (1,3,2) (0,0,2) (1,0,2) (1,0,1)
14  (0,0,0) (1,3,0) (0,3,0) (0,3,3) (1,3,3) (0,0,3) (1,0,3) (1,0,0)
15  (3,0,0) (2,3,0) (3,3,0) (3,3,3) (2,3,3) (3,0,3) (2,0,3) (2,0,0)
16  (0,1,0) (1,2,0) (0,2,0) (0,2,3) (1,2,3) (0,1,3) (1,1,3) (1,1,0)
17  # creating task on 64 PE
18  # sending parameter
19  # sending logical position to each cell
20  # sending logical mapping
21  # waiting stable
22  *** HOST_EXIT(0) ***
23  ----------------------------
24  Max: 1.819263 Min: 1.817908 Ave: 1.818579
```

マッピングパターン

【図27】

$4^3$を$8^2$にスキップ法でマッピングした場合の具体的結果を示す図

```
26  mapfile is map.skip.64
27  # 3 dimensional mapping
28  # 10000 bytes per a lattice
29  # 100 steps
30  # send-direction 6
31  # LOCELX= 4
32  # LOCELY= 4
33  # LOCELZ= 4
34  (0,0,0) (0,0,1) (0,1,0) (0,1,1) (0,2,0) (0,2,1) (0,3,0) (0,3,1)
35  (0,0,3) (0,0,2) (0,1,3) (0,1,2) (0,2,3) (0,2,2) (0,3,3) (0,3,2)
36  (1,0,0) (1,0,1) (1,1,0) (1,1,1) (1,2,0) (1,2,1) (1,3,0) (1,3,1)
37  (1,0,3) (1,0,2) (1,1,3) (1,1,2) (1,2,3) (1,2,2) (1,3,3) (1,3,2)
38  (2,0,0) (2,0,1) (2,1,0) (2,1,1) (2,2,0) (2,2,1) (2,3,0) (2,3,1)
39  (2,0,3) (2,0,2) (2,1,3) (2,1,2) (2,2,3) (2,2,2) (2,3,3) (2,3,2)
40  (3,0,0) (3,0,1) (3,1,0) (3,1,1) (3,2,0) (3,2,1) (3,3,0) (3,3,1)
41  (3,0,3) (3,0,2) (3,1,3) (3,1,2) (3,2,3) (3,2,2) (3,3,3) (3,3,2)
42  # creating task on 64 PE
43  # sending parameter
44  # sending logical position to each cell
45  # sending logical mapping
46  # waiting stable
47  *** HOST_EXIT(0) ***
48  ----------------------------
49  Max: 1.679813 Min: 1.679192 Ave: 1.679333
50  ----------------------------
```

【図28】

$4^3$ を $8^2$ にタイル法でマッピングした場合の具体的結果を示す図

```
52  # 3 dimensional mapping
53  # 10000 bytes per a lattice
54  # 100 steps
55  # send-direction 6
56  # LOCELX= 4
57  # LOCELY= 4
58  # LOCELZ= 4
59  (0,0,0) (0,1,0) (0,2,0) (0,3,0) (0,0,1) (0,1,1) (0,2,1) (0,3,1)
60  (1,0,0) (1,1,0) (1,2,0) (1,3,0) (1,0,1) (1,1,1) (1,2,1) (1,3,1)
61  (2,0,0) (2,1,0) (2,2,0) (2,3,0) (2,0,1) (2,1,1) (2,2,1) (2,3,1)
62  (3,0,0) (3,1,0) (3,2,0) (3,3,0) (3,0,1) (3,1,1) (3,2,1) (3,3,1)
63  (0,0,3) (0,1,3) (0,2,3) (0,3,3) (0,0,2) (0,1,2) (0,2,2) (0,3,2)
64  (1,0,3) (1,1,3) (1,2,3) (1,3,3) (1,0,2) (1,1,2) (1,2,2) (1,3,2)
65  (2,0,3) (2,1,3) (2,2,3) (2,3,3) (2,0,2) (2,1,2) (2,2,2) (2,3,2)
66  (3,0,3) (3,1,3) (3,2,3) (3,3,3) (3,0,2) (3,1,2) (3,2,2) (3,3,2)
67  # creating task on 64 PE
68  # sending parameter
69  # sending logical position to each cell
70  # sending logical mapping
71  # waiting stable
72  *** HOST_EXIT(0) ***
73  ----------------------------------------
74  Max: 1.887382 Min: 1.884576 Ave: 1.885689
75  ----------------------------------------
```

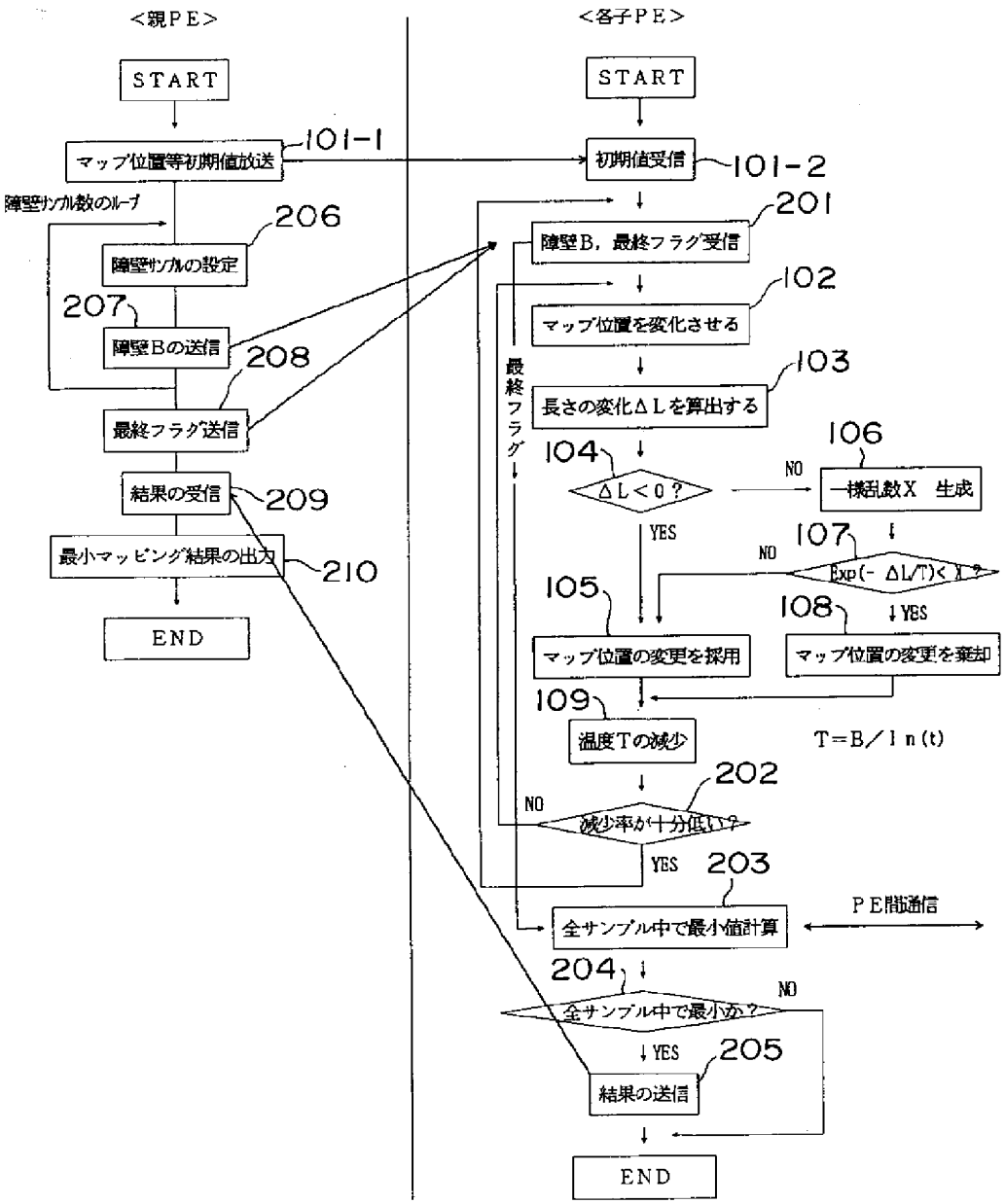
【図30】

最適マッピング情報データベースのファイル形式を示した図

1．取得情報種別（入力キー）
　1：ユニット分割次元数
　2：マッピングテーブル
2．計算モデル次元数：M（入力キー）
　1～10
3．データ参照関係：R（入力キー）
　1：最隣接PE
　2：近接　PE（斜め横を含む）
　3：全PE
　4以上：ユーザ指定
　　自PEからの間接アドレスで参照のあるPEを指定する。(0,-1),(1,-1)....
4．PE台数：N（入力キー）
　1～1024
5．ユニット単位の転送データ量：D（入力キー）
　　1～4：1～10000（バイト）
　5以上：ユーザ指定

6．ユニット分割次元数：UM（出力1）
　1～M－1
7．マッピングテーブル（出力2）
　論理PE番号　ブロックユニット番号

【図31】




---

フロントページの続き

(72)発明者　奥田　基
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内